

Adam Stephens Major project

by Adam Paul Stephens

FILE	18502_ADAM_PAUL_STEPHENS_ADAM_STEPHENS_MAJOR_PROJECT_78227_1634619610.DOCX (4.45M)		
TIME SUBMITTED	19-APR-2016 06:47 PM	WORD COUNT	16650
SUBMISSION ID	56073818	CHARACTER COUNT	89803



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

**Design and creation of a digital alarm clock with
A sunrise simulator,
And investigation of waking up to gradual light**

By
Adam Stephens

Supervisor: Dr. John Rees
Second marker: Dr. Gary Jones

Project submitted as part of the requirements for the
award of BEng (Hons) Computer systems and
electronics

April 2016

DECLARATION

I, Adam Paul Stephens declare that I am the sole author of this Project; that all references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

Signed :

Date :

FORM OF CONSENT

I Adam Paul Stephens hereby consent that my Project, submitted in candidature for the BEng (Hons) Computer Systems and Electronics degree, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations.

Signed :

Date :

Abstract

This report researches a theory that waking up to gradual light can motivate a person throughout the day and allow them to wake up feeling refreshed. To prove this theory dawn light simulators have been researched, and a microcontroller digital alarm clock with a dawn light simulator was designed. The design discusses different methods and implementations that can be used, once the final design has been decided the code required to operate the digital clock is programmed using C language. The final design is then created, and the code is implemented within the microcontroller, the prototype clock, and an existing dawn light simulator clock is given to participants, and a survey is used to gather data, the data is then analysed to draw a conclusion to prove the initial theory.

The report also discusses the future of the project by suggesting possible additions to the system and how testing can be improved to acquire accurate data for future trials.

Contents

Abstract	3
Table of figures	6
Chapter 1 - Introduction	8
Chapter 2 – Research and Review of literature	9
2.0: what is gradual light?	9
2.1: Why is gradual light important?	9
2.2: Existing products	11
2.2.1: What is LUX?	12
2.2.2: Lumie body clock GO 75	12
2.3: Research survey	14
2.4: Potential market	14
2.5: Research conclusion.....	14
Chapter 3 – Research methodology	15
3.0: Research onion	15
3.1.1: Layer 1: Positivism and interpetivism	16
3.1.2: Layer 2: Deductive	16
3.1.3: Layer 3: Experimental, Survey and Case study	17
3.1.4: Layer 4: Mixed methods	17
3.1.5: Layer 5: Longitudinal.....	17
3.1.6: Layer 6: Data collection and data analysis.....	18
3.2: Hardware and software required	18
Chapter 4 – Design & implementation	19
4.0: Initial design	19
4.1: Block diagram.....	20
4.2: Digital clock - Inputs.....	21
4.2.1: Buttons and switches.....	21
4.3: Digital clock - Outputs.....	24
4.3.1: Display – LCD & LED	24
4.3.2: Buzzer.....	28
4.3.3: Wireless communication	28
4.3.4: Dawn light simulation	30
4.4: Microcontroller selection.....	31
4.4.1: Timers	31

4.5: Final design	34
4.6: Implementation	34
4.6.1: counting time and displaying results	35
4.6.2: Changing the time	38
4.6.3: Setting the alarm and dawn light simulator alarm	40
4.6.4: Alarm, snooze and sunlight activation.....	41
4.6.5: Optional code.....	41
4.7: Creation of final design	42
4.7.1: Parts list.....	42
4.7.2: Testing the circuit	43
4.7.3: Circuit errors	46
4.8: Participant testing.....	47
Chapter 5 – Evaluation	48
5.1: Analysis of surveys	48
5.2: Future testing.....	48
5.3: Analysis conclusion	49
Chapter 6 – Conclusions & Recommendations	50
6.1: Conclusion	50
6.2: Future enhancement	50
Chapter 7 – Reflection.....	52
Chapter 8 - References	53
Chapter 9 - Bibliography	55
Appendices	56
Figure 1: LUX light level chart	56
Figure 2: survey and answers for initial research	57
Figure 3: Circuit diagram.....	59
Figure 4: PIC16F887 header. File	60
Figure 5: Flow chart for counting time and display	62
Figure 6: Flow chart for user changing time	63
Figure 7: Flow chart for user setting alarm and sunrise alarm	64
Figure 8: Flow chart for outputting alarm and sunrise simulator.....	65
Figure 9: PIC 16f887 code	66
Figure 10: Lumie body clock survey and answers.....	80
Figure 11: Prototype clock survey and answers	84

Table of figures

Figure 1: Product comparison chart	Page 11
Figure 2: Lumie body clock go 75	Page 12
Figure 3: Research onion	Page 15
Figure 4: Digital alarm clock block diagram	Page 20
Figure 5: Dawn light simulator block diagram	Page 20
Figure 6: Button open	Page 21
Figure 7: Button closed	Page 21
Figure 8: Example of button bouncing	Page 22
Figure 9: De-bouncing using components	Page 22
Figure 10: Switch open	Page 23
Figure 11: Switch closed	Page 23
Figure 12: LCD dot matrix	Page 24
Figure 13: Common anode 7 segment display	Page 25
Figure 14: Numbers for an LED display	Page 25
Figure 15: Segment required for 7 segment display	Page 26
Figure 16: Example of a 4 digit 7 segment display	Page 26
Figure 17: LED display example off	Page 27
Figure 18: LED display example on	Page 27
Figure 19: Buzzer example	Page 28
Figure 20: Quasar transmitter module QAM-TX2	Page 29
Figure 21: Quasar receiver module QAM-RX2	Page 29
Figure 22: Example of an LED connected to PIC	Page 30
Figure 23: PIC16F887 Timer 2	Page 31
Figure 24: PIC16F887 Pins	Page 32
Figure 25: Crystal oscillator example	Page 33
Figure 26: LM7805 Pins	Page 33
Figure 27: 5V Regulator example	Page 33
Figure 28: Seven segment display hex numbers	Page 35
Figure 29: Hex values defined	Page 35
Figure 30: 24 Hour counting code	Page 36

Figure 31: Sprintf statement	Page 36
Figure 32: Selecting the digit of the seven segment display	Page 37
Figure 33: Selecting the numbers of the seven segment display	Page 37
Figure 34: Simulated LED display working	Page 38
Figure 35: Code for changing time	Page 38
Figure 36: Minutes set to 59	Page 39
Figure 37: Minutes incrementing hours	Page 39
Figure 38: Hours set to 23	Page 39
Figure 39: Hours reset when equal to 24	Page 39
Figure 40: Selecting time displays	Page 40
Figure 41: Setting sunrise alarm	Page 40
Figure 42: Table of prices	Page 42
Figure 43: Batter supply voltage	Page 43
Figure 44: 5V Regulator measured	Page 43
Figure 45: Switch turned off example	Page 44
Figure 46: Switch turned on example	Page 44
Figure 47: LED display example off	Page 45
Figure 48: LED display example on	Page 45
Figure 49: LED supply voltage	Page 45
Figure 50: LED voltage drop	Page 45
Figure 51: Working circuit	Page 46

Chapter 1 - Introduction

Sleep is an important aspect of life it helps recover a person's body after a long day of various tasks, but the modern day methods of waking up in the mornings are flawed. As most people use standard alarm clocks or mobile phone alarms to start their day, this causes an immediate wake up which disturbs people from their sleep suddenly, which can lead to people feeling tired and demotivated throughout the day ahead.

The aims of this project are to prove that waking up to gradual light before an alarm activates will ease a person out of their sleep and enforce a more pleasant experienced wake-up, and allow that person to feel more refreshed and motivated throughout the day.

There are devices available on the market that promise to help a person wake up which will be investigated to determine what is required to test this theory.

To test this theory a few initial objectives need to be achieved:

- To design and create a microcontroller based digital alarm clock which a user can set the desired alarm.
- Develop and create a sunrise simulator paired with the digital clock which uses gradual light to simulate the sunrise.
- Testing the alarm clock on participants by allowing them to use the alarm clock to wake up, and by studying the effects using a survey.

By using participant's to test the consequences of a sunrise simulator and document their experience an unbiased conclusion can be gathered and compared to achieve a final result which will determine if the theory is correct.

Chapter 2 – Research and Review of Literature

To achieve a greater understanding of the areas needed to cover this project a wide range of research has been undertaken to ensure the most important aspects have been covered, this chapter will discuss the necessary parts of research needed to understand the project.

2.0: what is gradual light?

Gradual light is a form of light which is initially dark and progressively increases in intensity and brightness over time until the full brightness has been achieved. An example of gradual light is the sunrise, as the moon descends and the sun slowly rises at dawn the brightness increases over time until the sun is at its peak and the light remains at a constant level of intensity.

2.1: Why is gradual light important?

Dr. Natalie Dautovich a National Sleep Foundation's environmental scholar [1] states that "*your body's internal clock is sensitive to light and darkness,*" this internal clock is what allows the body to wake naturally up and be prepared for the day ahead. However, as modern life becomes busier people are forced to wake up earlier for demanding tasks such as work. And often wake up before sunrise especially in winter where the sun does not rise until approximately 7:30 am. due to this, many people suffer from a condition called SAD (Seasonal Affective Disorder) which is a form of depression brought on by the winter months. not much is known about SAD but it is commonly believed that is due to the lack of sunlight during the winter months, the National Health Service believe that:

The lack of sunlight might stop a part of the brain called the hypothalamus working properly, which may affect the:

- **production of Melatonin** – Melatonin is a hormone that makes you feel sleepy; in people with SAD, the body may produce it in higher than normal levels

- **production of serotonin** – Serotonin is a hormone that affects your mood, appetite and sleep; a lack of sunlight may lead to lower serotonin levels, which is linked to feelings of depression
- **body's internal clock (circadian rhythm)** –your body uses sunlight to time various important functions, such as when you wake up, so lower light levels during the winter may disrupt your body clock and lead to symptoms of SAD [2].

There are multiple believed cures for SAD such as medication and changes to a person's routine, but one of the most popular and efficient is a treatment called light therapy. In which a patient is given a device that emits bright light for a timed period which attempts to deceive the individual's brain into thinking there is more light within the day. However, this therapy requires the patient to be situated next to the device every morning for a time period between 30 minutes to an hour which most people cannot spare due to morning routines and daily commutes. It is also suggested that in some cases this treatment causes a condition called hypomania, this can cause the user to experience overactiveness, delusional states, low attention span and severely reduced required sleep, although this condition is rare the risks still outweighs the positive effects calling the need for a better solution.

A safer and similar cure for SAD uses a device called a dawn light simulator clock. This device is a digital alarm clock with a bright light implemented within, and the user is meant to set an alarm before they go to sleep. And usually, 30 minutes before the alarm will sound the light will turn on and gradually increase in brightness until the 30-minute duration is complete at which point the alarm will wake the user. This method of treatment simulates the sunrise which exposes the user to the required light while they sleep allowing the individual to still be able to perform their morning routines without interruptions.

There are multiple theories whether dawn light simulators are a more efficient treatment than light therapy or if the two methods make a difference to the user at all. One journal article conducted a study where 40 patients were exposed to two different devices, a dawn light simulator, and a light box and recorded the effectiveness of each, K.V. Danilenko and I.A. Ivanova concluded that [3] *"Dawn simulation reduced depression levels similarly to bright light in the treatment of winter depression. There was also a nearly equal distribution between those who would prefer dawn simulation and bright light for long-term use"*, from this statement it could be argued that both devices are just as effective, and a dawn light simulator is a suitable replacement for a light box.

However, due to the limited understanding of SAD and uncertainty of accuracy from study's conducted some researchers argue that simulated light does not have any effect.

One study who's aims were to disprove SAD's existence used a telephone study where 35,000 American citizens were called at random and asked to complete a survey asking multiple questions their about sleeping habits, the influence and effect the sun has in their opinion and if they feel depressed. Their research concludes that *"The researchers found these factors had no effect on the occurrence of depressive symptoms."*[4] But due to the unreliability of telephone surveys and the informal approach to the study methods the research conducted has a degree of uncertainty.

As seen, there are mixed theories on whether gradual light has any effect on a person or if the change of light can affect an individual's performance, research in this field must be critically analysed to determine if any changes in performance or health are noticed.

2.2: Existing products

There are multiple dawn light simulating products available on the market which claim to offer a suitable cure for SAD and the ability to enhance performance rates. By comparing the most popular products and observing their key features a general idea of how the simulator functions can be

<u>Product Features</u>				
Product name	Sunrise System light box	Lumie body clock GO 75	Philips wake-up light	Medisana Sunrise Alarm Clock
Display	LCD	LED	LED	LED
Light output (LUX)	10,000	10,000	200	n/a
Light type	LED	42W Bulb	LED	LED
Price	£109	£74.95	£95	£80.96

achieved, and a greater understanding of the requirements to build a dawn light simulator.

Figure 1: Product comparison chart.[5],[6],[7],[8]

Figure 1 shows the basic features of four popular dawn light simulating alarm clocks, each clock includes a feature where the gradual light will start 30 minutes before the alarm sounds, each clock also measures the light output in units of LUX.

2.2.1: What is LUX?

LUX is a unit used to measure light intensity from a light source onto the desired surface, for example, "1 lux is equal to the illumination of a surface one metre away from a single candle." [9] From this statement it can be said that most dawn light simulators claim to produce 10,000 LUX onto the user under the assumption the clock is used no more than 1 meter away. to measure the units of LUX a device called a LUX meter is used, according to a LUX chart [10] the sun on a clear day produces approximately 50,000 – 100,000 LUX.

Due to the vague descriptions of light output given by the dawn light simulators and lack of understanding of LUX measurements most people are sceptical of the brightness of light produced.

2.2.2: Lumie body clock GO 75



Figure 2: Lumie body clock GO 75

To achieve a higher understanding of a dawn light simulators and to acquire a greater in-depth knowledge of the requirements to create one, the Lumie body clock was purchased to test and analyse, and to be used on participants to gather data.

2.2.2.1: How does it work?

The Lumie body clock has a compact design which incorporates the clock within the base of the product, and a 42W bulb located within the top encased in a plastic shell. The user sets the time and the desired alarm for wake up, 30 minutes before the alarm is scheduled to sound the bulb will activate and begin to increase in brightness gradually for the time period, once the 30 minutes has passed the alarm sounds and wakes the user.

2.2.2.2: Observations and assumptions

By using the dawn light simulator for a period of time a few assumptions and comments can be noted.

The plastic shell is used to change the colour of the light produced from bright white to a warm yellow for simulating the sun more accurately.

The clock has been designed to use minimal space by only using three buttons and an LED display as seen on the faceplate. The buttons are used for multiple features such as setting time, setting the alarm, snooze, changing the alarm sound and turning the light on or off. Due to the simplistic design users could find it complex to operate the clock without referring to the user manual each time, this could cause some features of the clock never being used and effect the marketing of the product.

The clock is powered by a mains supply so it can be assumed due to mains power having a frequency of 60Hz, that the frequency is used to generate the clocks time base where 60Hz is equal to 1 second.

The clock also includes a feature which when the system is disconnected from the power source the time will remain correct for 1 hour. It is assumed that within the system is a small microcontroller which is constantly updated with the time, and if the systems power is disconnected this microcontroller is powered by a single cell battery and continues counting the time for 1 hour, the battery is also assumed to be charging when the system is connected to mains.

The overall design of the clock is well implemented but does retain a few flaws such as the bulb, due to its characteristics a bulb generates a lot of heat and has a low life expectancy requiring replacements to be often bought. By replacing this with multiple LED lights, the same results can be achieved while increasing life expectancy and also reduce cost prices.

2.2.2.3: Lumie body clock conclusion

The system is well designed and compact but by changing some features, the clock can be optimised allowing a simpler use and a reduction of cost to build the product.

2.3: Research survey

To achieve a better understanding of people's morning routines and how they wake up on a daily basis, an online survey was created using Survey Monkey[11] and participants were encouraged to answer a short series of questions anonymously. 51 people completed the questionnaire allowing for a more in-depth understanding of how the consumer base operates and what they would like to see within a product (Questionnaire and answers in appendix figure 2).

The survey provided a greater understanding of the potential market of consumers and what is necessary to include within the design for a successful product.

2.4: Potential market

The main marketing of dawn light simulators is people who suffer from SAD, but as the research survey shows the majority of people wake up at 8 am or earlier, at this time often the sun has not risen yet especially in the winter months. The survey also suggests that most people feel irritable and demotivated after waking up, by applying the information to the product a new area of marketing can be achieved, which is necessary due to the survey also stating that the majority of people have not heard of a dawn light simulator.

2.5: Research conclusion

The research shows there is a need within the market for a dawn light simulator which proves that waking up to gradual light will allow people to wake up refreshed, less irritable and show an increase in motivation. The product also needs to be a cost effective solution as the research survey shows that most people are willing to pay between £10 - £20 and that no one is willing to pay more than £40 for such a product. This highlights the need for a cheaper solution within the market as all current products are in excess of £50 onwards.

Chapter 3 – Research Methodology

Research methodology is used to gather data for the product using multiple methods, a common example used is called the research onion where the methods are broken down into layers, and a path is used to get the desired results.

3.0: Research onion

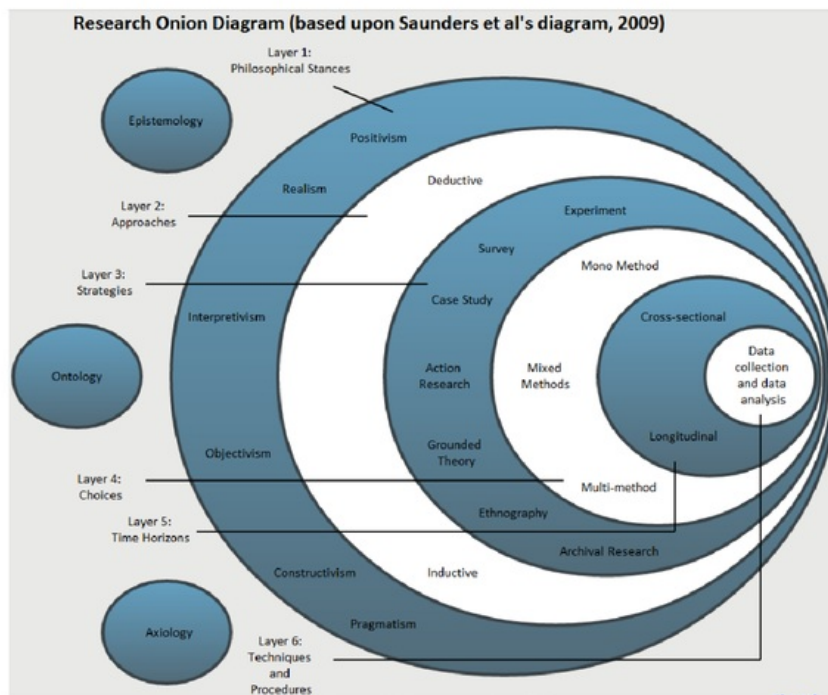


Figure 3: Research onion [12]

Figure 3 represents the research onion where each layer contains the methods required to achieve data; these methods are broken down into five layers:

- Layer 1: Philosophical stances
- Layer 2: Approaches
- Layer 3: Strategies
- Layer 4: Choices
- Layer 5: Time horizons

And the sixth layer contains all of the previous methods used to gather the data.

Each layer of the research onion was studied carefully, and the appropriate methods were used to collect the required data. There are three methods to approach the research onion Epistemology, Ontology, and Axiology. The approach chosen is Epistemology due to the research using a scientific approach with definite answers; the following methods were used to gather data.

3.1.1: Layer 1: Positivism and Interpretivism

Layer 1 is called the philosophical stance which can be perceived as the structure for the research methods, as this layer defines what the outcome of the data will be by stating what type of research is required.

Positivism

Positivism is a method used to generate questions for research where a problem is gathered from existing work within the field of investigation, is method is chosen as multiple case studies argue different results from testing gradual light on subjects, this requires a definitive answer to be generated but also taking into consideration other researchers work.

Interpretivism

Interpretivism generates data by gathering results from people's experiences within the researchers work, is method is required due to testing of the product being dependent on participants comments and experiences using the product.

Both these methods are necessary to gather information needed to determine that the product is successful and answers the necessary questions to draw a conclusion of the problem.

3.1.2: Layer 2: Deductive

Layer 2 states how the research approaches generating the required information; a deductive approach is required due to beginning the research with a general theory to be answered. Then progressively narrowing information gathered from testing to draw a conclusion that answers the question initially stated.

3.1.3: Layer 3: Experimental, Survey and Case Study

Layer 3 contains the different strategies for gathering data which are then carefully examined to output a relevant conclusion for the initial theory.

Experimental

The experimental strategy gathers data by a scientific method where answers are collected and observed to output a conclusion for the problem; this strategy is used to achieve data from a group of participants and conclude the results.

Survey

The survey strategy involves asking participants to answer desired questions and gather data from the grouped responses. This method is a vital part of the research where a questionnaire is issued on the internet to gather primary research for the initial problem and also used to generate a conclusion from testing the product on individuals and studying the responses.

Case study

The case study strategy is used on people to gather real term answers, this method is primarily used in conjunction with the survey strategy, due to questioning real people to gather practical information for the desired problem.

Using these three strategies data is ensure to be precise and able to give relevant information to conclude the theory.

3.1.4: Layer 4: Mixed methods

Layer 4 suggests how the data is acquired by either a quantitative or qualitative method; mixed methods use both quantitative and qualitative to ensure data gathered is accurate and allows for both methods to output a single conclusion.

3.1.5: Layer 5: Longitudinal

Layer 5 is used to state the period at which data is gathered; the longitudinal method is chosen as data collected requires participants to answer questions presented throughout their day.

3.1.6: Layer 6: Data collection and data analysis

Layer 6 contains all previous methods implement to gather data, and from here the data is analysed to achieve the conclusion of the theory.

As the methods of the research, onion shows each layer is necessary to gather appropriate data relating to the theory and demonstrating the approach required, once a clear path is shown the methods can be implemented, and the data can be processed to output a clear conclusion of the problem.

3.2: Hardware and software required

To design and create the product various hardware and software is necessary for simulation and creation.

Software

- Proteus V8 simulation
- PCW CCS C Compiler
- MPLAB
- Easy PC
- Yed graph editor

Hardware

- Breadboard
- Soldering iron
- (Black box)
- Multimeter
- Pick-it 3
- Digimess MI200 (All-in-one instrument)
- Oscilloscope

Listed above are the most important instruments required for project completion, not every listed product may be used and some devices not listed may be necessary.

Chapter 4 – Design & Implementation

This chapter of the report covers the design process by breaking down each section into understandable parts and discusses the possible components and methods required to create a prototype and also shows how each part operates by explaining the necessary code.

4.0: Initial design

The initial design of a project states the requirements that the designer wishes to include within their product, this allows for a greater understanding of which components are necessary to output a completed product.

The initial project product idea is to create a digital alarm clock that communicates wirelessly to a dawn light simulator. For the design to be created each section is broken down into three main parts, the clock, the dawn light simulator and the wireless communication, each part then states the required functions for operation.

Digital alarm clock requirement

- To be able to count 24 hours to simulate a day cycle accurately and display the results.
- For the user to be able to set the time of the device.
- To allow the user to set a desired alarm and display what time the alarm is set for.
- Once the alarm is activated, generate a sound from a buzzer until the alarm is switched off.
- Incorporate a snooze button where if the alarm is activated the alarm will deactivate for a time period before reactivating.

Dawn light simulator requirements

- To produce gradual light to simulate the sunrise 30 minutes before the alarm is activated.

Wireless communication requirements

- The clock must transmit a signal 30 minutes before the alarm is activated, to the receiver on the dawn light simulator to initiate the start of the dawn light simulator.

The system is required to use a microcontroller due to the functions that the product is expected to perform, a microcontroller acts as the brain of any system by achieving data from inputs, performing any functions required within the code which are stored in the chip, and outputting data in the required form.

4.1: Block diagram

By observing the system requirements stated previously, information can be acquired from each section of the design; this information can then be processed to produce a basic block diagram of the overall system.

Digital alarm clock blocks diagram



Figure 4: Digital alarm clock block diagram

Dawn light simulator block diagram

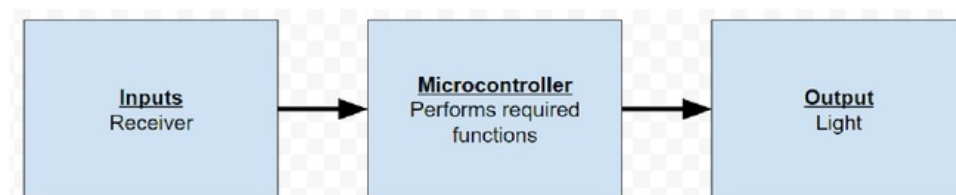


Figure 5: Dawnlight simulator block diagram

The block diagrams above show the basic requirements necessary to design and create a functional system, each part of the block diagram is broken down into segments and different solutions and methods to the problems are investigated; this allows a wide area of possible design to be covered.

Before an appropriate microcontroller can be selected the number of inputs and output must be established for the pin count and the required function of the microcontroller must be identified.

4.2: Digital clock - Inputs

4.2.1: Buttons and switches

From the requirements stated above for the system, different information can be gathered. The clock requires three buttons for incrementing hours, incrementing minutes and for the snooze function, the system also requires three switches for setting the time, setting the alarm and one for turning the alarm on and off. The switches and buttons are the inputs required for the microcontroller and are fundamental to operate the clock.

By using proteus simulation, the design of the inputs can be achieved, and an understanding of how they operate is acquired.

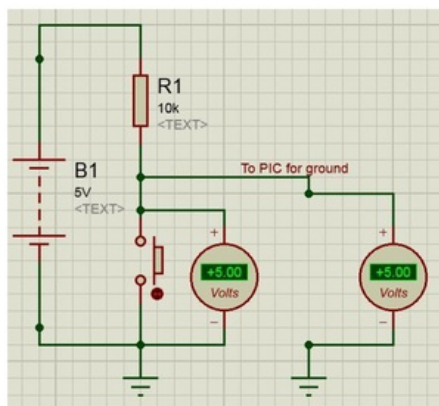


Figure 6: Button open

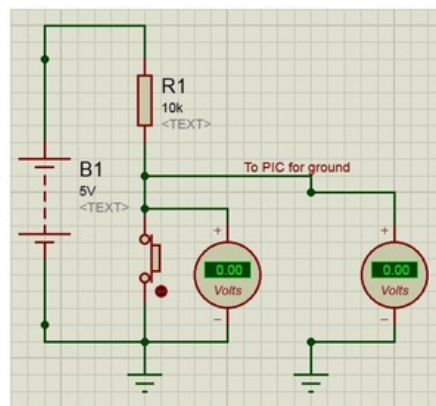


Figure 7: Button closed

The images above show how the buttons operate, by connecting one side of the button to ground and the other to the 5V power supply using a pull-up resistor; a wire can be attached from the microcontroller input pin which is internally connected to ground. As figure 6 shows when the button is not pressed 5V is passed through the microcontroller and grounded making the pin input high (5V). However, as figure 7 demonstrates once the button has been pressed a connection is created effectively shorting power to ground. And passes 0V to the microcontroller input stating that the pin is low (0V). from this it can be said that when that when the input to the microcontroller is low the required function can be performed.

Due to the characteristics of a button, once the button has been pressed an effect called bouncing occurs. This effect causes the voltage to rapidly increase and decrease between 5V and 0V when the button is pressed and released. And due to the microcontroller depending on values being high (5V) or low (0V) the system is unable to tell if the button has truly been press or how many times.

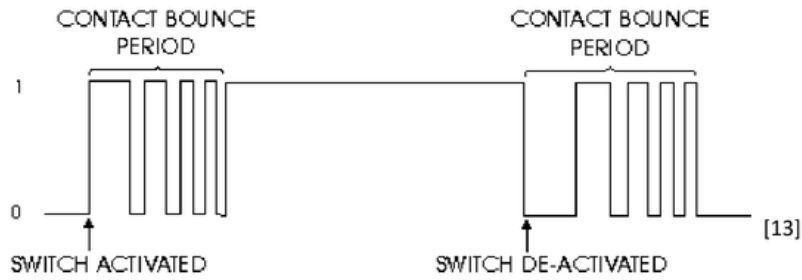


Figure 8: Example of button bouncing

Figure 8 shows a case of bouncing where even though the button has only been pressed once the voltage pulses multiple times, and the microcontroller registers each pulse as a press and performs the function. This gives unreliable data due to if a button is pressed once to add 1 minute to the current time the microcontroller could register that the button has been pressed ten times instead.

To solve this problem a method called de-bouncing is required, there are two main methods a button can be debounced.

Component de-bouncing

One method involves adding a resistor and capacitor to the button to steadily increase and decrease the voltage level when the button is pressed.

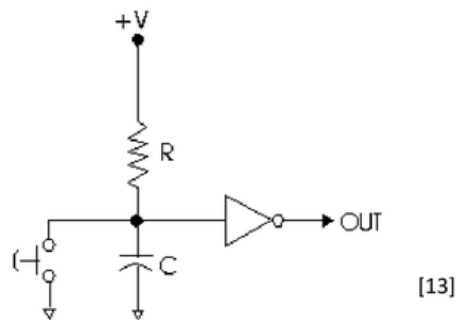


Figure 9: De-bouncing using components

Figure 9 shows a resistor and capacitor used to de-bounce the voltage; the capacitor holds a charge and when the button is pressed the charge is released at a steady rate until the button is released. And which point the capacitor charge will increase at a steady pace, the charge and discharge of the capacitor bypass the bouncing time, a buffer can also be introduced to allow for a cleaner signal.

The disadvantages of this method are that it requires extra components which take up room within the system and increase the design cost.

Software debouncing

Another method of de-bouncing can be implemented within the code of the microcontroller, by adding a short delay of a few milliseconds when the button is pressed and another delay when the button is released, this forces the microcontroller to ignore the changes in voltage for the delay period effectively skipping the voltage bouncing.

This method is used within the design due to requiring fewer components allowing for the design to be smaller and more cost effective.

The product also requires three switches for setting the time, the alarm and turning the alarm on and off; this is because when these functions are used the system requires them to be on for the period of time they are to be used.

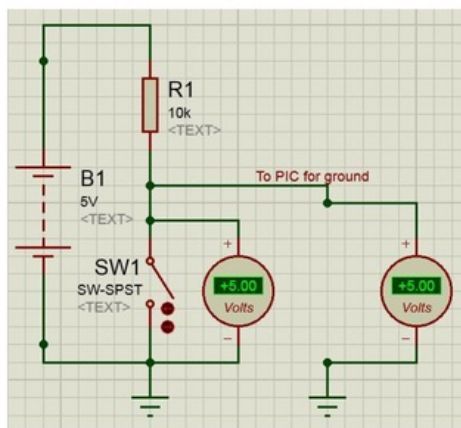


Figure 10: Switch open

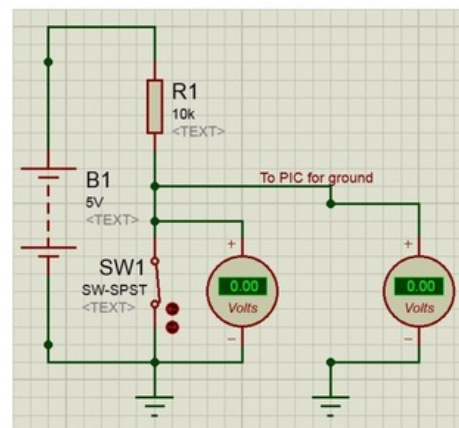


Figure 11: Switch closed

As figure 10 and Figure 11 show the switches use the same setup as the buttons where a high input to the microcontroller represents the switch is off, and a low input indicates the switch is on, the only difference is the push button is replaced with a single latch switch.

4.3: Digital clock - Outputs

4.3.1: Display – LCD & LED

The display of a digital alarm clock is one of the most important parts as it is used to show the time and alarm of the system. The two main types of display are LCD (Liquid Crystal Display) and the LED (Light Emitting Diode) display, both of these displays have advantages and disadvantages and will determine how the system will function.

Liquid Crystal Display

An LCD is a commonly used device in electronics as it allows multiple characters and numbers to be displayed easily and has a low power consumption. The most widely used LCD is a 2x16 where there are two rows, and each row has a possible 16 characters that can be displayed, these characters each have their rows and columns which are used to show the required number or letter.

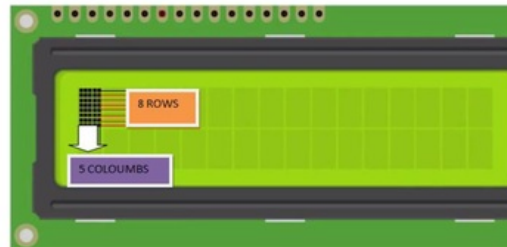


Figure 12: LCD dot matrix [14]

Figure 12 shows an example of how a single character functions within a 2x16 LCD, this shows a 5x8 character within the display on, which is referred to as a dot matrix. A standard LCD will have 16 pins, 8 of which are data input pins, and these are used to select which parts of each character are on and then shift to the next character. There are also two pins for an LED backlight which is the anode and cathode; these can be used in conjunction with a variable resistor to adjust how bright the backlight is for the LCD.

The main benefits of an LCD are that it allows multiple characters to be shown on the system.

LED display

An LED display commonly referred to as a seven segment display uses eight light emitting diodes to display selected characters and numbers, these displays are often found within digital clock due to being able to easily display numbers.

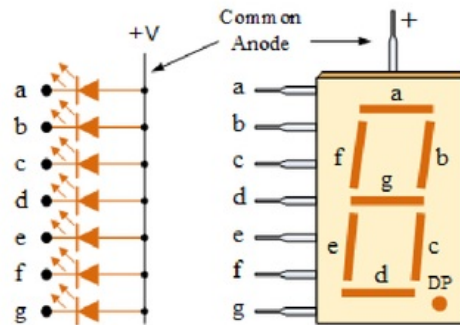


Figure 13: Common anode 7 segment display [15]

There are two types of 7 segment displays, the common anode display where power is applied to the display and each segment required to be on is grounded as shown in figure 13, and the common cathode where power is being implemented to each segment required to be on and they share a pin connected to ground.

As figure 13 shows each segment of the display consists of its own light emitting diode labelled a – g and have their individual pins to allow them to turn on for the desired number.

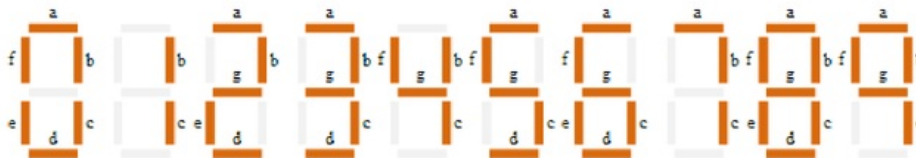


Figure 14: Numbers for an LED display [15]

Figure 14 shows each possible number that can be displayed and which segments need to be turned on to display them, as a microcontroller only operates output value high or low the requirements for each desired digit must be defined.

Number / letter	Segment A	Segment B	Segment C	Segment D	Segment E	Segment F	Segment G	Segment DP
Number 0	0	0	0	0	0	0	1	1
Number 1	1	0	0	1	1	1	1	1
Number 2	0	0	1	0	0	1	0	1
Number 3	0	0	0	0	1	1	0	1
Number 4	1	0	0	1	1	0	0	1
Number 5	0	1	0	0	1	0	0	1
Number 6	0	1	0	0	0	0	0	1
Number 7	0	0	0	1	1	1	1	1
Number 8	0	0	0	0	0	0	0	1
Number 9	0	0	0	0	1	0	0	1
Letter A	0	0	0	1	0	0	0	1
Letter B	1	1	0	0	0	0	0	1
Letter C	0	1	1	0	0	0	1	1
Letter D	1	0	0	0	0	1	0	1
Letter E	0	1	1	0	0	0	0	1
Letter F	0	1	1	1	0	0	0	1

Figure 15: Segments required for 7 segment display

Figure 15 shows the required output from the microcontroller pins to display different numbers and letters where 0 is low (Ground), and 1 is high (Power in). Number 9 also uses segment D for its display unlike figure 14 shows as this was found to represent the number more efficiently, letters are also included for possible future uses.

For the screen of a 24-hour clock, four 7 segment displays are required to show hours and minutes, but due to display only being able to show one digit at a time a technique called multiplexing is required. This is where only 1 7 segment is selected at a time, and the required segments are lit to show that digit, the next segment is then lit, and the process continues for all four segments continuously. Because of the high speed that this occurs the human eye registers all the displays on at once showing their individual numbers.



Figure 16: Example of 4 digit 7 segment display [16]

Figure 16 shows a four-digit 7 segment which has 12 pins; four pins are used to supply power to each digit, and the remaining 8 are used for the segments.

In order to multiplex the display, a PNP transistor is required for each digit of the screen.

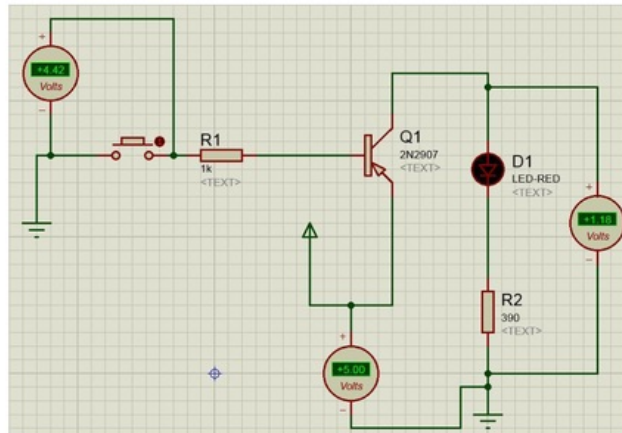


Figure 17: LED display example off

Figure 17 shows an example of how each digit is connected to the display. LED (D1) is used to represent a digit of the display, and the button is used to represent the microcontroller. A 2N2907 PNP transistor is used where the base is connected to the PIC with a 1k current limiting resistor; the emitter is directly linked to 5v, and the collector is attached to the anode of the display. The PNP transistor allows voltage to flow from the emitter to the base this is used when the digit of the display is required to be off, but when the base of the transistor is grounded the voltage from the emitter is allowed to flow directly to the collector as shown in figure 18.

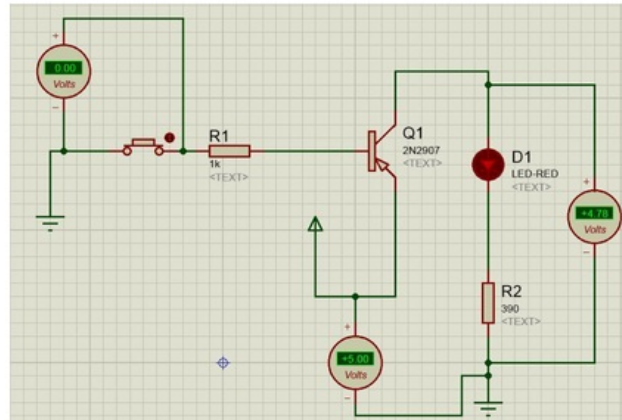


Figure 18: LED display example on

As the digits share the anode to power the display the individual segments must be grounded to show the desired value, this is done by switching the pins of the microcontroller where the segment that is required to be on is low for ground, and the segments that need to be off are high.

The LED display has been selected for the design as it is cheaper, easier to use and fulfils the requirement of the clock where only 24 hours need to be displayed before resetting.

4.3.2: Buzzer

A buzzer is used as an alarm within the product to wake the user, for this a simple self-activated piezo buzzer is implemented where one wire of the buzzer is connected to the power supply, and the other is connected to a pin of the microcontroller as shown below.

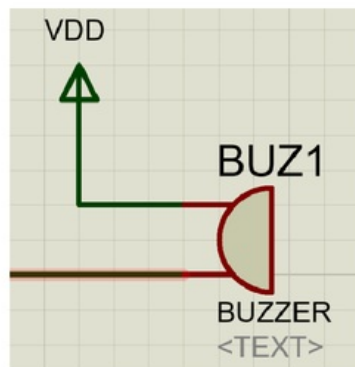


Figure 19: Buzzer example

When the output of the microcontroller is high, the buzzer is off, but when the output is low (ground), the buzzer is activated.

4.3.3: Wireless communication

The initial design of the system will use wireless communication where a transmitter is connected to the digital clock circuit, and a receiver is connected to the dawn light simulator circuit, 30 minutes before that alarm of the clock is activated data is to be sent to the receiver circuit to initiate the sunrise simulator.

There are many different types of wireless communication the many two types that can be used for the project are IR (infrared) and RF (Radio Frequency). Infra-Red uses ambient light set to a desired frequency commonly using a 555 timer chip, but due to the dawn light simulator producing bright light this could potentially interfere with the wireless signal.

RF communication uses radio waves within the MHz range and requires paired modules due to laws restricting certain frequencies.

RF communication can be implemented in multiple forms; the most common are using a paired transmitter and receiver module using UART through the utilization of a microcontrollers TX (transmit) and RX (receive) pins.

Transmitter

The transmitter module is used to send serial data to the receiver to allow the dawn light simulator activate the lighting process.

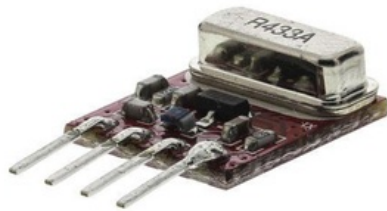


Figure 20: Quasar Transmitter module QAM-TX2 [17]

Figure 20 shows the Quasar transmitter module which operates at a frequency of 433 MHz. The module has four pins used for different functions, where one pin is used if the system requires an external antenna, one pin is connected to ground, another is used for power supply and the last is used as a data input pin.

Receiver

The receiver module is required to operate at the same frequency as the transmitter module which is 433 MHz, and is used to acquire any signal the transmitter outputs.

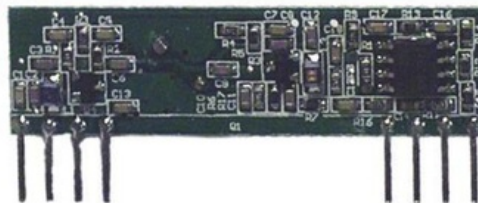


Figure 21: Quasar Receiver module QAM-RX2 [18]

Figure 21 shows the Quasar receiver module, which contains eight pins, where one pin can be used for an external antenna, three pins are used for ground, two pins for the supply voltage and two pins are used for the data input.

Due to time constraints, the wireless communication was removed from the final design and a contingency plan was enforced where the lights for the dawn light simulator were added into the clock circuit.

4.3.4: Dawnlight simulation

The dawn light simulator requires multiple lights to gradually increase in brightness for a period of 30 minutes. The lights are required to be bright enough to simulate the sunrise and ease the user out of their sleep. Rather than use a bulb which is implemented with the Lumie body clock, LED's are chosen as they have a broad range of colour, and brightness can be adjusted by adding or removing individual LED's for the desired lighting. LED's also consume less power and have a higher life span than bulbs.

To gradually increase the LED's brightness 1 LED is turned on at set intervals over a 30 minute time period, this is implemented by connecting the cathode of the LED to ground and the anode to a pin of the microcontroller through a current limiting resistor as represented in figure 22.

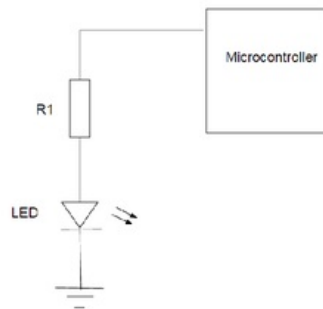


Figure 22: Example of an LED connected to PIC

In order to achieve the required brightness multiple LED's were tested where it was found that bright white was the most efficient in simulating the sun, and the most effective within this category was the Cree C503D-WAN-CCbEb151 [19]. By referring to the datasheet, it states that the LED requires a voltage drop of approximately 3.2V and a typical current of 20mA, from a supply voltage of 5V where the necessary voltage to the LED is 1.8, and the desired current is 20mA the resistance can be calculated using the following formula.

$$RESISTANCE = \frac{VOLTAGE\ DROP}{REQUIRED\ CURRENT}$$

$$RESISTANCE = \frac{1.8V}{20mA}$$

This gives a resistor value of 90 Ohms, but when tested this was too bright to notice a change in light, so a 120 Ohm was implemented. Through testing different numbers of LED's, it was found that 6 supplied and efficient brightness, from this it can be said that 1 LED is required to be turned on every 5 minutes to allow for gradual lighting.

4.4: Microcontroller selection

For the desired use of the system, a microchip 8 bit PIC is chosen due to the low requirements and the cost effectiveness.

From the above sections, it can be determined that the microcontroller requires a minimum 25 input /output pins for each connection required, the microcontroller also requires a feature called a timer.

4.4.1: Timers

Timers are commonly used within microcontrollers where time is a high dependency, the digital clock requires a timer in order to count the time continuously, timers use a clock frequency to determine time base.

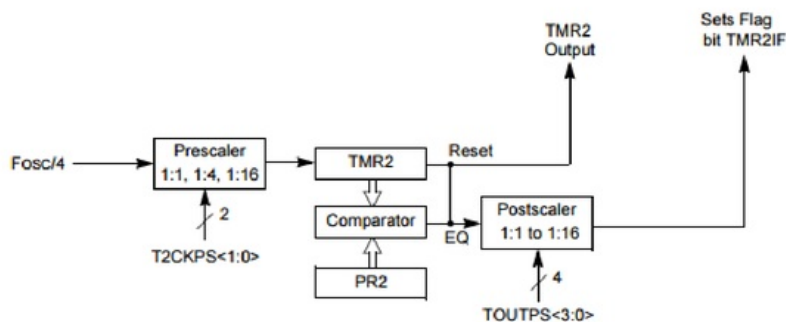


Figure 23: PIC 16f887 Timer 2 [20]

Figure 23 shows an example of a timer where F_{osc} is the clock input, this is divided by 4 when it enters which mean a clock frequency of 4MHz is only 1Mhz when it enters the timer, this is then passed through a Prescaler which defines what time period the timer uses.

Using the 4 MHz clock frequency sets the microcontroller to perform 1 million instructions per second and by setting the timers to prescaler to 4 allows for an interrupt to be performed every one millisecond, this can then be used to change the time accurately.

By using microchip's PIC selector[21] for the microcontroller's requirements, two chips were found that are suitable for the design, the PIC16F886, and the PIC16F887. These two chips have the same capability but have a different pin amount where the PIC16F886 has 25 i/o pins, and the PIC16F887 has 35 i/o pins. As the system requires a minimum of 25 i/o pins, the PIC16F887 is selected to ensure there are enough pins and to accommodate for future enhancements.

PIC16F887 key features

Program Memory:	14
RAM:	368 Bytes
Data EEPROM	256 Bytes
Timers:	2 x 8-bit, 1 x 16-bit
ADC:	14 ch, 10-bit

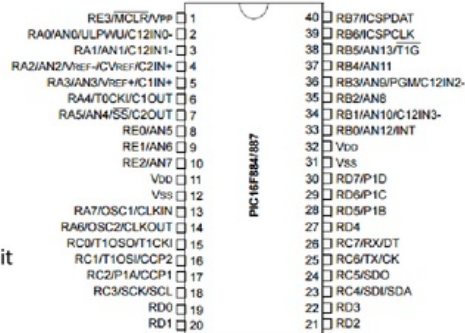


Figure 24: PIC16F887 pins [20]

Temperature Range (C):	-40 to 125
Operating Voltage Range (V):	2 to 5.5
Pin Count:	40

The features of the PIC16F887 are acquired from the microchip datasheet[20].

The microcontroller requires an oscillator to apply a clock frequency for the circuit to operate; the microcontroller contains an internal oscillator that can operate between 8 MHz and 32 MHz. But due to previous use, a 4 MHz crystal oscillator is used which is connected to pins 13 and 14 for oscillation as shown in figure 24.

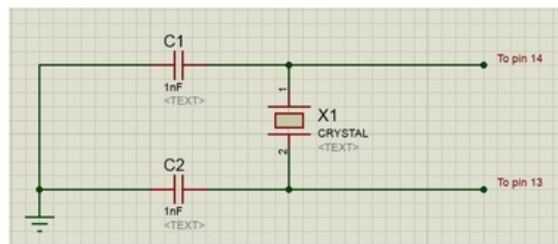


Figure 25: Crystal oscillator example

Figure 25 shows the typical arrangement for an external crystal oscillator where X1 is the 4 MHz crystal and C1 & C2 are used to smooth out the oscillating signal.

The PIC microcontroller has an operating voltage of 2 – 5.5V, and the other components requiring 5V; this shows that a voltage regulator must be implemented within the product to reduce the battery supply, for this an LM7805 voltage regulator is chosen. Which contains three pins input, ground, and output as shown in figure 26

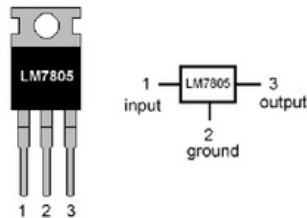


Figure 26: LM7805 pins [22]

The input is the supply voltage from the battery; the ground is 0v, and the output is a regulated 5V, the voltage regulator operates more efficiently with lower input voltages due to higher voltages producing significant amounts of heat. Therefore, a 6V power supply consisting of 4 AA battery's is ideal for the input.

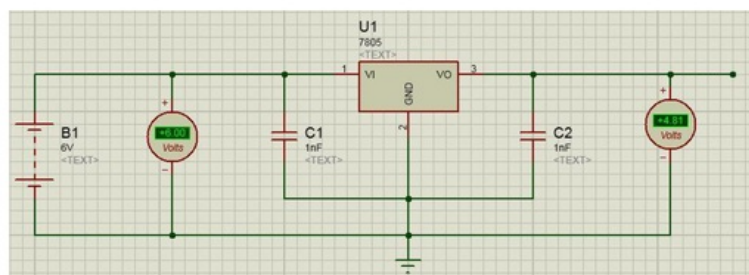


Figure 27: 5V regulator example

Figure 27 shows how a voltage regulator is connected. It can be seen that a 6V supply is applied to the input of the regulator, and a voltage of 4.8V is produced. As this is close enough to the required circuit supply, the 6V battery can be used, capacitors C1 and C2 are used to decouple the voltages to allow for a cleaner supply.

4.5: Final design

Once all components have been decided the final design can be created using an electronics circuit simulator (Proteus). The final design consists of the PIC16F887 microcontroller, three buttons, and three switches which are utilized for the inputs, a four digit 7 segment display, a piezo buzzer for the alarm and 6 LED's to simulate the sunrise (for full circuit diagram refer to figure 3 in appendices).

4.6: Implementation

The implementation section is used to demonstrate how the coding of the microcontroller works and how the physical product is constructed. To understand how the code works each section is broken down into individual functions, the programming was written using C coding due to more experience using this language; a CCS C compiler is used with Proteus to initially set up the PIC and code.

Using the CCS C compiler, the requirements of the system were defined, and a header file was created (See appendices figure 4 for full header file). The header file contains and applies the settings for the pic such as a brownout timer where if the microcontroller voltage drops below 2.1V the system resets. The file also contains the pin names whereby defining a name to a pin this can be used to select the pin allowing for easier coding to that pin without referring to the circuit diagram. As all the pins on port B are used for the LED display segments they can simply be defined as B.

For the display segments, the required values as seen in figure 15 are converted from binary into Hexadecimal as shown below.

Number/Letter	HEX value
0	03
1	9F
2	25
3	0d
4	99
5	49
6	41
7	1F
8	01
9	09
A	11
B	C1
C	63
D	85
E	61
F	71

Figure 28: 7 segment display hex numbers

```

#define num0 0x03
#define num1 0x9f
#define num2 0x25
#define num3 0x0d
#define num4 0x99
#define num5 0x49
#define num6 0x41
#define num7 0x1f
#define num8 0x01
#define num9 0x09
#define numa 0x11
#define numb 0xc1
#define numc 0x63
#define numd 0x85
#define nume 0x61
#define numf 0x71

```

Figure 29: hex values defined

These values are then determined in the header file as shown in figure 29 where after define appears the digit is given a name, and 0x is used to state the following values are hexadecimal which is then added.

After the header file has been created the code can then be implemented within the source file, flow charts are also used to break the code down to understand what is required.

4.6.1: counting time and displaying results

The digital clock requires an accurate time keeping and to be able to view the time on the 7 segment display. The system uses timer 2 to allow an interrupt every one millisecond which can be used to count up to 24 hours. A flow chart of the process is created to define what each section is required to perform (see figure 5 in appendices for flowchart). The code requires four integers to be created, 'Val' which contains the milliseconds, 'secs' to count how many seconds have passed, 'minutes' to count how many minutes have passed and 'hours' to count how many hours have passed.

By allowing the timer interrupt to increment 'Val' by 1 each time the interrupt is called this creates a millisecond counter. Using an if statement allows for conditions to be followed by the code when the integer 'Val' is equal to 1000 (1 second) the integer 'secs' is incremented by 1 and 'Val' is reset. This indicates that 1 second has passed and the millisecond counter has reset and begins incrementing again. This process is repeated throughout the code where when 60 seconds have passed the integer 'minutes' is incremented by 1 and 'secs' is reset, when 60 minutes have passed the integer 'hours' is incremented by 1 'minutes' is reset. And when 'hours' is equal to 24 'hours' and 'minutes'

is reset, this enables to the system to accurately count to 24 hours for a day cycle before resetting and starting the count again.

```
#INT_TIMER2
void TIMER2_isr(void) // Timer 2 interrupt (1mS)
{
    val++; // millisecond counter
    if(val==1000) //second counter
    {
        secs++; // used to increment seconds
        val=0; // reset millisecond
        if(secs==60) // minute counter
        {
            minutes++; // used to increment minutes
            secs=0; // resets seconds
            if(minutes==60) // hour counter
            {
                hours++; // used to increment hours
                minutes=0; // resets minutes
                if(hours==24) // day counter
                {
                    hours=0; // resets hours
                    minutes=0; // resets minutes
                }
            }
        }
    }
}
```

Figure 30: 24-hour counting code

Figure 30 shows how the logic is implemented within the code. Once the system can count 24 hours it is required to display the results to the four-digit 7 segment display, this requires converting the values of the integers from decimal to hexadecimal and for each character to be individually displayed to the correct digit of the display. This is achieved by creating a char (Character) called 'message' and using a sprintf function to output the value of 'hours' and 'minutes' as a string to allows each number to be used individually.

```
sprintf(message,"%02u%02u",hours,minutes);
```

Figure 31: Sprintf statement

Figure 31 shows the sprintf statement where 'hours' is converted into hex and the two digits are placed into the array 'message' as a string. Then 'minutes' uses the same process to display the numbers to the correct digit of the 7 segment display. Each digit must be turned on 1 at a time; this is done using a for loop where each segment is powered from port d of the PIC.

```

for (q=0;q<4;q++) // loop for displaying LED multiplexing showing 1 7segment at a time with a value
{
    a=message[5-q]; // use a to store 1 part of an array to display
    delay_ms(10); // 1 ms delay allow the LED 7 segment display to be seen (to be displayed on pin5seg higher value is needed)
    output_d(0x0ff); // Puts all PORTB high ready for output (Multiplexing)
    switch(q) // use q as a variable to display select 7 segment
    {
        case 0: output_low(PIN_D3); // used to select Pin D3 low allowing
                break; // 7 segment for hours (ten) to be displayed
        case 1: output_low(PIN_D2); // used to select Pin D2 low allowing
                break; // 7 segment for hours (units) to be displayed
        case 2: output_low(PIN_D1); // used to select Pin D1 low allowing
                break; // 7 segment for minutes (ten) to be displayed
        case 3: output_low(PIN_D0); // used to select Pin D0 low allowing
                break; // 7 segment for minutes (unit) to be displayed
        default: break;
    }
}

```

Figure 32: Selecting the digits of the 7 segment display

Figure 32 shows the code used to cycle which digit of the display is powered, where an integer q is used to loop four times for each segment. The pins are required to be high before they are selected this resets the process each loop. A switch case is used as an if statement where there are multiple conditions that can be met. The selected microcontroller pin is then powered low to turn on the display digit, and a break is used to exit the switch statement.

Once the digit of the display is selected the correct number needs to be displayed, this is achieved by adding another switch case statement to the code which is used to choose the output for the number required.

```

output_b(0x0ff); // used to set PORTB high ready for outputing the 7 segment value
switch(a)
{
    case '0': output_b(num0); // output 0 on led display
                break;
    case '1': output_b(num1); // output 1 on led display
                break;
    case '2': output_b(num2); // output 2 on led display
                break;
    case '3': output_b(num3); // output 3 on led display
                break;
    case '4': output_b(num4); // output 4 on led display
                break;
    case '5': output_b(num5); // output 5 on led display
                break;
    case '6': output_b(num6); // output 6 on led display
                break;
    case '7': output_b(num7); // output 7 on led display
                break;
    case '8': output_b(num8); // output 8 on led display
                break;
    case '9': output_b(num9); // output 9 on led display
                break;
    case 'A': output_b(numa); // output A on led display
                break;
    case 'B': output_b(numb); // output B on led display
                break;
    case 'C': output_b(numc); // output C on led display
                break;
    case 'D': output_b(numd); // output D on led display
                break;
    case 'E': output_b(nume); // output E on led display
                break;
    case 'F': output_b(numf); // output F on led display
                break;
    default : break;
}
if(q==0) output_low(PIN_B0); // set PIN B0 low for Decimal Place for Hours unit 7 segment

```

Figure 33: Selecting the numbers for the 7 segment display

Figure 33 shows integer 'a' which contains the value for the display is used to select which number is required. Each case uses the ASCII value and when 'a' matches that value that case is performed where required PORT B pins become low for an output. Due to the pin values previously defined within the header file, the names given can be used instead of entering each pin. an if statement is also used for digit 3 of the display to turn on the decimal point segment to easily the separation between hours and minutes. As seen below an example of the simulated display is achieved using four random numbers.



Figure 34: Simulated LED display working

Due to the speed that the microcontroller produces the numbers a delay was required to slow down the display digits so the values could be seen by the human eye.

4.6.2: Changing the time

The digital clock requires the user to be able to set the desired time. For the user to be able to set the time a switch called set time is required to be on when the switch is on two buttons 'minutes' and 'hours' are needed to increment the corresponding values. If the value exceeds their desired limit, a condition is enforced to rectify the value (flow chart shown in figure 6 in appendices).

```
// set the time loop
if(!input(set_time)) // When pin C2 is low (set time switch) use this statement
{
    if(!input(hrs)) // If Pin C5 (Hours button pressed) use this statement
    {
        delay_ms(500); // Delay for debouncing
        hours++; // incrementing hours
        delay_ms(500); // Delay for debouncing
        if(hours==24) // if the value hours is 24 use this statement
        {
            hours=0; // reset hours
        }
    }
    if(!input(mins)) // If Pin C4 (Minutes button pressed) use this statement
    {
        delay_ms(500); // Delay for debouncing
        minutes++; // incrementing minutes
        delay_ms(500); // Delay for debouncing
        if(minutes==60) //if the value minutes is 60 use this statement
        {
            hours++; // Hour increment
            minutes=0; // resets minutes to 0
        }
    }
}
// end of setting time
```

Figure 35: Code for changing time

Figure 35 shows the code used to allow the user to modify the time, an if statement is used for a condition where if the set time input pin is low (switch is turned on) and if the hours or minutes button is pressed increment hours or minutes by 1. a delay is used to debounce the inputs for when the button is pressed, if statements are also used for the value of minutes where if minutes exceeds 60 hours is incremented by 1 and minutes is reset. And hours if statement condition if hours exceed 24 then hours is reset, this eliminates the chances of setting the values higher than desired. Once the set time pin turns high (switch is turned off) the code then skips this statement until the time is set by the user again.



Figure 36: Minutes set to 59



Figure 37: Minutes incrementing hours

The figures above demonstrate when minutes is equal to 59 and the button is pressed once more the minutes value resets and hours is incremented by 1.



Figure 38: Hours set to 23



Figure 39: Hours reset when equal to 24

Figure 38 and 39 also show that when hours is equal to 23 and the hours button is used to increment the hours again hours resets, from the simulation it can be seen that the code is operating as desired and any unwanted value errors are conditioned for.

4.6.3: Setting the alarm and dawn light simulator alarm

The digital clock requires the user to set desired alarm for waking up, and the system sets the sunrise alarm for 30 minutes before the alarm time. To achieve this once the set alarm pin is low (set alarm switch on), the clock needs to output the alarm time on the 7 segment display. The same methods can then be used as when the user sets the time. Once the alarm is set the system needs to subtract 30 minutes from the alarm to set when the dawn light simulator is activated.

To show the alarm time on the 7 segment display two new integers are made called 'alarm_hours' and 'alarm_minutes,' a second sprintf statement is then created and by using an if statement for when the set alarm switch is activated the correct display can be seen.

```
if(!input(set_alar)) //if PIN_C1 is 0 use this statement for alarm display
{
    sprintf(message,"%02u%02u",alarm_hours,alarm_minutes);    // convert decimal to hex for alarm
}
else // other wise use this statement for time display
{
    sprintf(message,"%02u%02u",hours,minutes);                // convert decimal to hex for time
}
```

Figure 40: selecting time displays

Once the system displays the alarm time the same method used to set the time can be implemented but using the integer for the alarm.

After the user has set the alarm, the system is required to subtract 30 minutes from the alarm and store the time to initiate the sunrise sequence. This is achieved by creating two integers called 'sunrise_hours' and 'sunrise_minutes' which are then set to the time of the alarm. A third integer is then created called 'Sunrise' which have a value of 30 stored within, a for loop can then be used with the value 'Sunrise' to subtract 30 minutes as shown below.

```
sunrise_hours = alarm_hours; // sets the sunrise hours to alarm hours
sunrise_minutes = alarm_minutes; // sets the sunrise minutes to alarm minutes

int sunrise = 30; // sunrise is used to loop 30 minute count down

for(sunrise = 30; sunrise != 0; sunrise --)
{
    if(sunrise_minutes == 0) // if sunrise minutes is equal to 0
    {
        sunrise_hours = sunrise_hours - 1; // subtract 1 from sunrise hours
        sunrise_minutes = 59; // set sunrise minutes to 59
    }

    sunrise_minutes = sunrise_minutes -1; // subtract 1 from sunrise minutes
}

// end of setting alarm and sunrise alarm
```

Figure 41: setting the sunrise alarm

An if statement is also used to condition the alarm where if the value of sunrise minutes is equal to 0 1 hour is subtracted and the minutes is reset to 59, this solves any errors when the alarm is being set (Flowchart is shown in figure 7 in appendices).

4.6.4: Alarm, snooze, and sunlight activation

Once the time and alarm have been set the system is required to activate 6 LED's each one at 5-minute intervals 30 minutes before the alarm. This is achieved using an if statement where if the sunrise alarm time matches the current time, and that alarm on / off switch is on then the statement is entered. Within the statement an integer called 'LED' which is pre-set to 0 is used to control multiple if statements, when the condition is met an LED will turn on, and a for loop is used to increment the next sunrise alarm by 5 allowing for 5 minute intervals and the integer 'LED' is incremented by 1. for the next LED in the sequence (Flowchart shown in figure 8 in appendices).

To activate the alarm a similar if statement is used where if the alarm time matches the current time an integer called 'alarm' is set to 1 to represent the alarm is active. This is then used to condition an if statement where if the alarm is equal to 1 the buzzer pin turns low activating the buzzer.

A snooze button is also used where if the button is pressed a for loop adds 5 minutes to the alarm and turns off the buzzer allowing a 5-minute interval before the alarm sounds again.

Once the alarm is desired to be shut down the user is required to turn off the alarm on / off switch, once this is done to pin is set high, and an if statement shuts off the alarm on the lights.

4.6.5: Optional code

Extra code was added to the product due to noting when the set alarm and set time switch were not in use the hours button and minutes button have no function. The code was added where if the hours button was pressed and neither the set time or set alarm switch were active all the LED's light up and a second press would turn off the LED'S allowing the product also to be used as a light source.

A similar design was implemented for the minutes button where if the button was pressed while set the alarm and set time switch were not activated 1 LED lights up each press of the button, once all 6 LED's are lit a final press of the button turns them all off.

4.7: Creation of final design

Once the circuit diagram has been designed and the code has been created and tested the final physical product can be build using breadboard as a prototype. Once the prototype is created, tests can be performed to ensure the design is operating correctly, and the product can be given to participants for testing to analyse and draw a conclusion.

4.7.1: Parts list

The components used to create the final project, and their prices are listed below:

Components	Quantity	Price
PIC16F887	1	£1.83
Kingbright 57 – 0194 LED display	1	£3.09
2N2907A PNP Transistors	4	£6.08
LM7805 Voltage regulator	1	£0.41
Piezo buzzer	1	£2.00
4Mhz crystal oscillator	1	£0.97
Cree white LED	6	£1.56
Push button	3	£5.56
SPST switch	3	£2.94
10k Ohm resistor	10	£2.80
1K Ohm resistor	4	£0.60
120 Ohm resistor	6	£1.62
33pF Capacitor	2	£0.72
1nF Capacitor	3	£0.81
Total:		£30.99

Figure 42: Table of prices

All components priced as single purchase's and selected from RS Online [23].

4.7.2: Testing the circuit

Once the components are acquired the product can be built and tested, this section shows the physical circuit constructed and measures the results; these are then compared to the simulated result shown within the design section to ensure the circuit is operating correctly.

Voltage regulator



Figure 43: Battery supply voltage

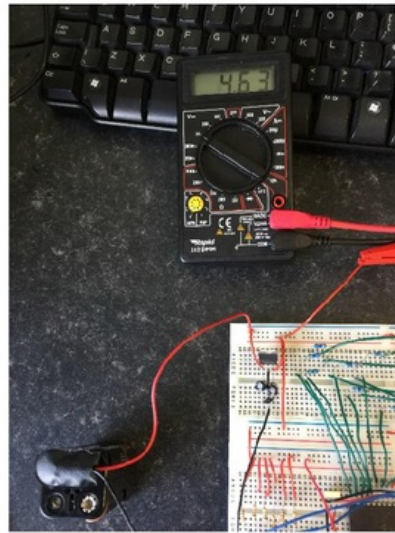


Figure 44: 5V regulator measured

Figure 43 shows 4 AA battery's connected in series producing a voltage of 6.2 which is to be supplied to a 5V regulator as seen in figure 44. It can be seen that the output from the voltage regulator supply's 4.6V for the circuit, comparing this to the estimated 4.8V seen within the design section this is appropriate for the circuit.

This supply is also used to power both the input pins of the PIC16F887 where a decoupling capacitor is connected between the supply and ground to eliminate any unwanted interference.

Buttons and Switches

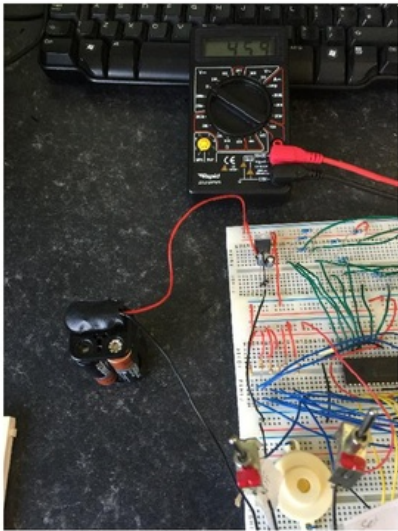


Figure 45: Switch turned off example

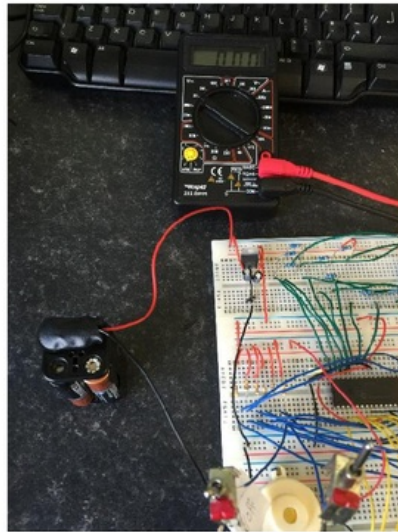


Figure 46: Switch turned on example

Figure 45 shows an example where when a switch is not activated 4.5V is supplied to the input pin of the microcontroller stating that the input is high when the switch is activated as figure 46 demonstrates the supply is grounded through the switch setting the input pin low to perform the required function.

This method is used for all of the buttons and switches used as inputs for the microcontroller.

Seven segment display example

Due to the high speed that a seven segment display operates it is difficult to measure the voltage but using an LED instead allows the circuit to be demonstrated.

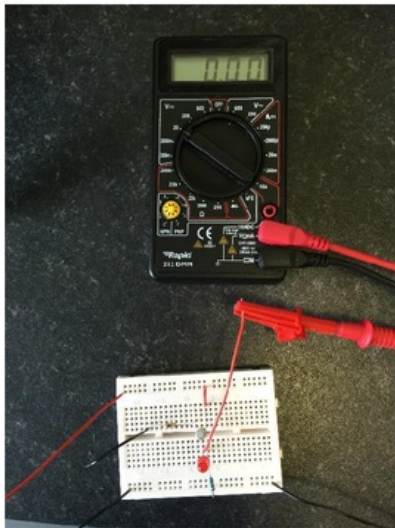


Figure 47: LED display example off

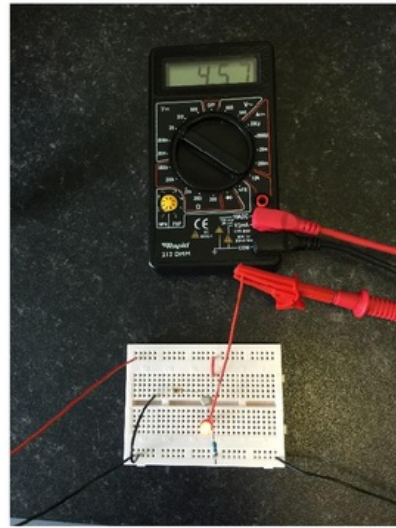


Figure 48: LED display example on

The above images represent how the 7 segment display operates where the transistor supplies the power to each digit, and the LED represents one segment. Figure 47 shows the transmitter base is not connected therefore the collector connected to the LED has no voltage. However, when the base is grounded voltage flows from the emitter to the collector powering the LED as shown in figure 48.

Sunrise simulator LED's

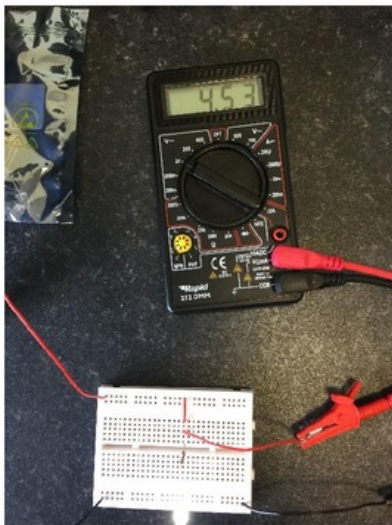


Figure 49: LED supply voltage

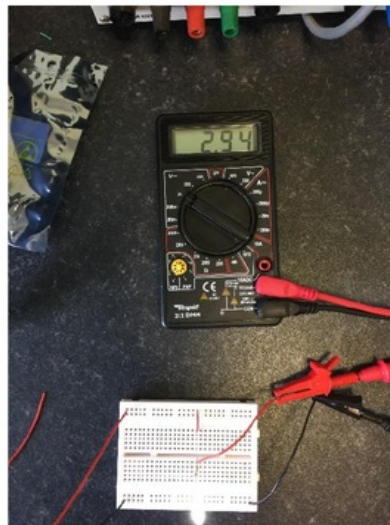


Figure 50: LED voltage drop

The images above demonstrate how the sunrise simulator operates where the voltage is supplied from the microcontroller when the output pin becomes high the connected LED turns on.

Once the microcontroller code and the circuits is operating as required the code is applied to the PIC, this is achieved using a PICKIT 3 and MPLAB. When the microcontroller was applied to the circuit, an error was discovered where the display was causing a flickering effect and segments were not switching on when required. It was determined the errors were caused by the delays within the code, due to Proteus simulation not running in real time the delays needed to be reduced the accommodate for the real-time system. The modified code was then added to the PIC, and the system was fully operational.

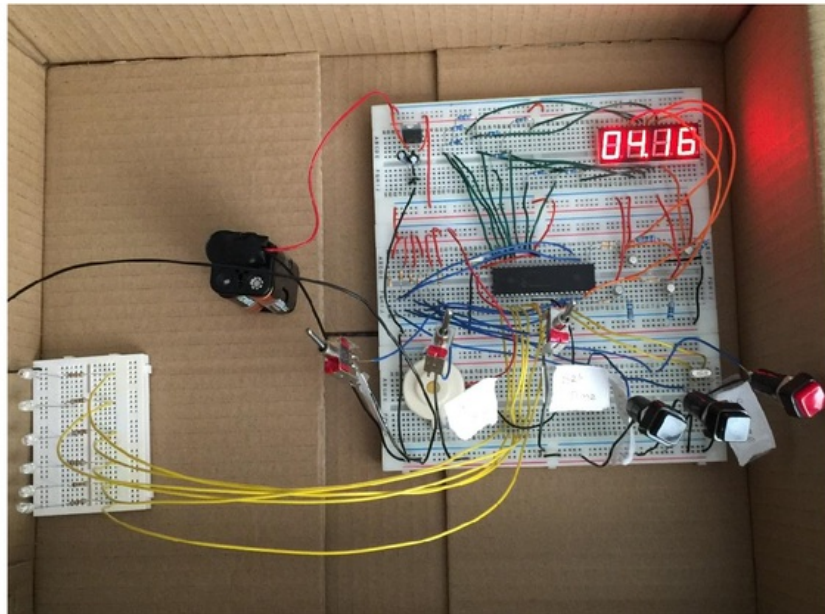


Figure 51: Working circuit

4.7.3: Circuit errors

Some errors are present within the circuit but due to time constraints could not be removed from the final design but have been noted, and solutions have been considered.

The LED display has a mild ghosting effect where the digits show a slight residue from the previous digit; this can be resolved but adjusting the delay within the multiplexing code.

Occasionally when setting the time or alarm and the hours button is pressed the all the sunrise LED turn on, this can be solved by adding another condition to the if statement which turns the lights on when the switches are not in use.

If the minutes button is pressed to turn on the LED's one at a time and during the sequence the hours button is pressed to turn all lights on and off. When the minutes button is pressed again, the sequence will continue from the last LED on, this can be solved by resetting the value of the LED's when the hours button is pressed.

4.8: Participant testing

In order to gather data on the effects of waking up to gradual light 5 participants were given two products the Lumie body clock Go and the prototype clock and instructed to use each for waking up. Two surveys were then given to the participant's one for the Lumie clock (Survey questions and answers in appendices figure 10) and another for the prototype clock (see appendices figure 11). To ensure each test was fair the participant were informed as little as possible about both clocks and the study. And to achieve fair and accurate result the Lumie body clock surveys were used as control testing.

Chapter 5 – Evaluation

This chapter investigates the results of the surveys used to test the clocks given and analyse the data to draw a conclusion; the section also discusses how the survey would be issued if another test were to be performed.

5.1: Analysis of Surveys

From the survey results, it can be seen that both clocks are effective at waking the individual, and both show that the majority of participant felt more refreshed and motivated throughout the day. An interesting comment from one participant this that even though the sun was up when they woke, it was still dark due to the British weather. This information shows potential for the sunrise simulator to be used all year round and not just during the winter months. Another unexpected answer is that even though the users did not feel the LED's simulated the sun they were preferred over the Lumie clocks bulb.

Easy to use was one of the key features implemented for the prototype clock. And as the surveys indicate the participants found the clock simple to use, for the Lumie body clock one user suggested they would like to see a snooze button implemented. As there is a snooze feature, this enforces the statement that the Lumie clock's small design complicates the use, and some features are not used.

From the surveys, it can be seen that none of the users would be willing to pay more than £50 for either product. As the Lumie body clock costs £75 this severely affects the marketing of the device, but as the components of the prototype clock only cost £30 it can be said that the prototype clock is a cost effective solution.

The surveys can be analysed to say that the initial theory has been confirmed where the majority of users state that they felt refreshed waking up and were motivated throughout the day, this shows that there is a correlation between waking up to gradual light and how this impacts people's behaviour throughout the day.

5.2: Future testing

For future testing of this product, a few requirement need to be added to ensure a fair and detailed test.

- Control testing: by using a third survey which asks questions about the participant's normal sleeping habit's a general idea of how the individual behaves can be achieved, this can then be compared to the final test to draw a conclusion.
- More participants: by using more people to test the products a wider are of data can be achieved to give a definite answer to the theory.
- Longer testing periods: by allowing the participants to use each clock for longer periods such as a week the final conclusion can be considered more accurate, this is due to better testing where the user's behaviour could be different of the weather could change.

Due to time constraints these method could not be included but if added to future testing a definite and accurate result could be drawn.

5.3: Analysis conclusion

From the results, It can be said that the theory was correct where gradual light in the morning can enhance a person's motivation and prepare them for the day ahead, but due to the limited testing caused by time constraints a second testing with the previous methods included would confirm the results.

Chapter 6 – Conclusions & Recommendations

This section overviews how successful the project was and if the overall goal was achieved, and also discusses how the product can be improved in the future.

6.1: Conclusion

The project as a whole was a success due to a product being created and testing the outcome, although the wireless capabilities were never included the initial theory which states gradual light is able to motivate a person throughout the day and allow them to wake up refreshed, this requires more testing to give a definite answer it is believed to overall theory has been confirmed.

6.2: Future enhancement

A key part of any product's lifespan is investigating what can be changed or added to improve the product; this section discusses what ideas could be used to improve the system.

- **Wireless:** by adding the wireless capability which is discussed in the design section the dawn light simulator can be moved to desired locations, this could be used to simulate the sun better by placing the simulator by a window.
- **LCD screen:** an LCD screen could be used instead of the LED display, this would allow the user to see information about the clock such as time, alarm time, if the alarm is on and the date.
- **Internal oscillation:** instead of a crystal oscillator the microcontrollers own internal oscillator could be used, this would reduce the size of the board required and lower the price.
- **Radio and sounds:** A key feature that the participant's state they would like to see included within the system is a wide choice of sound, as the sound of the alarm is what wakes people up a calmer sound may help the individual wake up more refreshed, a participant also suggested they would like a radio feature this would allow a different alarm every morning.
- **Power supply:** by replacing the battery supply with a mains power supply this would ensure the clock will remain on without the worry or cost of replacing the batteries, this would require a transformer to be implemented to convert the mains ac supply to dc and lower the voltage.
- **Time keeping:** by adding a small microcontroller within the system which stores the current time and powered by a small single cell battery, if the product is disconnected from the power supply the

system can remember the current time for as long as the battery lasts, the battery can also be charge using the main power source.

- Mobile app: the initial survey shows that 88% of people use a mobile phone as their alarm which is a broad area of the market. By creating the wireless dawn light simulator and pairing it to a mobile app via Bluetooth the clock can be removed, this would allow compatibility with a wider market and servery reduce the product cost.
- PWM: by including PWM (Pulse Width Modulation) to the dawn light simulator the LED's can slowly fade on. This would allow for a better simulation of the sun.

Chapter 7 – Reflection

A lot of information has been acquired throughout the project which was not initially known, this section covers what key skills have been acquired throughout the project duration.

- Time management: time is a key factor in any aspect of life but through the major project time had to be carefully planned to complete required section within the designated time period.
- Programming: before the project only basic C programming had been used, but due to using a microcontroller C language had been researched and put to a physical use to code the PIC for the required functions, this increased personal knowledge and understanding of how a microcontroller operates and created an interest in microcontroller based projects.
- Knowledge of electronics: through researching the design of the product and observing different designs and method, personal knowledge of electronics and how the operate has increased and a better approach to designing a circuit has been achieved.
- Microcontroller's importance: as the project required a PIC to operate the various functions research had to be made within the PIC microcontrollers, from this research additional knowledge of how a PIC microcontroller operates and what the system include, this knowledge can be used for future project and designs.

The overall project was a success where multiple skills were acquired, and a higher understanding of how electronics operates was acquired, due to time constraints the project was not completed as initially planned but a working prototype had been created, if the project had extra time it is believed a lot more could be achieved.

The project as a whole was an enjoyable experience due to requiring all knowledge learned in previous years required to complete; this has inspired further researching to gain a broad area of knowledge within the electronics field for future work and small projects.

Chapter 8 - References

- [1] A. Medaris and N. Dautovich, "7 reasons you're waking up all wrong," *Huffpost healthy living*, 2015. [Online]. Available: http://www.huffingtonpost.com/2015/01/27/waking-up-mistakes_n_6547278.html. [Accessed: 28-Mar-2016].
- [2] NHS choices, "Seasonal affective disorder (SAD) - NHS Choices," 2015. [Online]. Available: <http://www.nhs.uk/Conditions/Seasonal-affective-disorder/Pages/Introduction.aspx>. [Accessed: 04-Apr-2016].
- [3] I. . Danilenko, K.V Ivanova, "Dawn simulation vs. bright light in seasonal affective disorder," *J. Affect. Disord.*, vol. 180, pp. 87–89, 2015.
- [4] N. Choices, "Seasonal affective disorder 'may be a myth', study argues - Health News - NHS Choices," 2016. [Online]. Available: <http://www.nhs.uk/news/2016/01January/Pages/Seasonal-affective-disorder-may-be-a-myth-study-argues.aspx>. [Accessed: 13-Apr-2016].
- [5] "Sunrise system SRS 320," *Bodykind*, 2016. [Online]. Available: <http://www.bodykind.com/search/sunrise-system-300-light/0/0/0/0/0/0/0/0/0/0/5/12/0/product/504-Sunrise-System-SRS-320-Sunrise-Alarm-Clock.aspx>. [Accessed: 13-Apr-2016].
- [6] "Lumie Bodyclock GO 75," *Lumie*, 2016. [Online]. Available: <http://www.lumie.com/collections/wake-up-lights/products/bodyclock-go-75>. [Accessed: 13-Apr-2016].
- [7] "Philips wake-up light," *PHILIPS*, 2016. [Online]. Available: http://www.philips.co.uk/c-p/HF3505_01/wake-up-light/overview. [Accessed: 13-Apr-2016].
- [8] "Medisana sunrise alarm clock," *bodykind*, 2016. [Online]. Available: http://www.bodykind.com/product/5362-Medisana-Sunrise-Alarm-Clock-WL-450-Wake_Up-Light.aspx#description. [Accessed: 13-Apr-2016].
- [9] "LUX, Lumens & watts," *GREEN BUSINESS LIGHT*, 2016. [Online]. Available: <http://www.greenbusinesslight.com/page/119/lux-lumens-and-watts>. [Accessed: 13-Apr-2016].
- [10] "LUX Light level chart," *use-IP Ltd.* [Online]. Available: https://www.use-ip.co.uk/datasheets/lux_light_level_chart.pdf. [Accessed: 13-Apr-2016].
- [11] "SurveyMonkey," *SurveyMonkey*, 2016. [Online]. Available: https://www.surveymonkey.com/?ut_source=header. [Accessed: 13-Apr-2016].
- [12] S. et al, "Research Onion diagram," *University of Derby*, 2016. [Online]. Available: <https://onion.derby.ac.uk/>. [Accessed: 13-Apr-2016].
- [13] "Contact Bounce and De-Bouncing," *Electronix Express*, 2016. [Online]. Available: http://www.elexp.com/contact_bounce_and_debounce.aspx. [Accessed: 12-Apr-2016].
- [14] "Character lcd working," *Microcontroller projects*, 2012. [Online]. Available: <http://www.microcontroller-project.com/16x2-lcd-working.html>. [Accessed: 13-Apr-2016].
- [15] "7-segment Display," *Electronics Tutorials*, 2016. [Online]. Available: <http://www.electronics-tutorials.ws/blog/7-segment-display-tutorial.html>. [Accessed: 13-Apr-2016].

- [16] "Kingbright 4 digit 7 segment display," *RS online*, 2016. .
- [17] "Quasar Telemetry module QAM-TX2," *RS online*, 2016. [Online]. Available: <http://uk.rs-online.com/web/p/lower-power-rf-modules/7620080/>. [Accessed: 14-Apr-2016].
- [18] "Quasar Telemetry module QAM-RX2," *RS online*, 2016. [Online]. Available: <http://uk.rs-online.com/web/p/lower-power-rf-modules/6172044/>. [Accessed: 14-Apr-2016].
- [19] "Cree white LED," *RS online*, 2016. [Online]. Available: <http://uk.rs-online.com/web/p/visible-leds/8106705/>. [Accessed: 14-Apr-2016].
- [20] "PIC16F882/883/884/886/887," *Microchip*, 2006. [Online]. Available: <http://ww1.microchip.com/downloads/en/DeviceDoc/41291G.pdf>. [Accessed: 14-Apr-2016].
- [21] "All 8-bit Microcontrollers Products," *Microchip*, 2016. [Online]. Available: <http://www.microchip.com/ParamChartSearch/Chart.aspx?branchID=1012>. [Accessed: 14-Apr-2016].
- [22] M. Shenoy, "5V Power supply using 7805 voltage regulator with design," *electroSome*. [Online]. Available: <https://electrosome.com/power-supply-design-5v-7805-voltage-regulator/>. [Accessed: 14-Apr-2016].
- [23] "RS online," *RS online*, 2016. [Online]. Available: <http://uk.rs-online.com/web/>. [Accessed: 15-Apr-2016].

Chapter 9 - Bibliography

- [1] P. Horowitz and W. Hill, *The art of electronics*, third edit. Cambridge university press, 2015.
- [2] R. Williams, *Real-time systems development*. Buttersworth-Heinemann publications, 2006.
- [3] M. P. Bates, *Programming 8-BIT PIC microcontrollers in C*. Newnes publication, 2008.
- [4] R. Barnett, L. O'Cull, and S. Cox, *Embedded C programming and the microchip PIC*. Thomson learning, 2004.
- [5] D. T. MESTROVIC, "Light therapy safety and side effects," *News medical*, 2013. [Online]. Available: <http://www.news-medical.net/health/Light-Therapy-Safety-and-Side-Effects.aspx>. [Accessed: 18-Apr-2016].
- [6] K. Darton and B. Murphy, "Understanding mania and hypomania," *Mind*, 2011. [Online]. Available: <http://www.nhs.uk/ipgmedia/National/Mind/assets/Understandinghypomaniaandmania.pdf>. [Accessed: 18-Apr-2016].
- [7] M. K. Traffanstedt, S. Mehta, and S. G. LoBello, "Major Depression With Seasonal Variation: Is It a Valid Construct?," *Clin. Psychol. Sci.*, p. 2167702615615867, Jan. 2016.
- [8] S. et al, "Research Onion diagram," *University of Derby*, 2016. [Online]. Available: <https://onion.derby.ac.uk/>. [Accessed: 13-Apr-2016].
- [9] "Character lcd working," *Microcontroller projects*, 2012. [Online]. Available: <http://www.microcontroller-project.com/16x2-lcd-working.html>. [Accessed: 13-Apr-2016].
- [10] "PIC16F882/883/884/886/887," *Microchip*, 2006. [Online]. Available: <http://ww1.microchip.com/downloads/en/DeviceDoc/41291G.pdf>. [Accessed: 14-Apr-2016].
- [11] M. T. J. Juan and S. Terman, "Light Therapy."

Appendices

Figure 1: LUX light level chart

Lux Light Level Chart

	LUX	DESCRIPTION
	50,000	British summer sunshine
	5,000	Overcast sky
	500	Well-lit office
	300	Minimum for easy reading
	50	Passageway/outside working area
	15	Good main road lighting
	10	Sunset
	5	Typical side road lighting
	2	Minimum security risk lighting
	1	Twilight
	0.3	Clear full moon
	0.1	Typical moonlight/cloudy sky
	0.001	Typical starlight
	0.0001	Poor starlight

[10]

Figure 2: survey and answers for initial research

How do you wake up? Total of 51 people answered.

1) What device do you use to wake up?

Alarm clock	6%	3 people
Mobile phone	88%	45 people
Other:	6%	3 people
None	2 people, 1 iPad	

2) On average what time do you wake up?

Before 6am	14%	7 people
Between 6am and 7am	31%	16 people
Between 7am and 8am	29%	15 people
Between 8am and 9am	16%	8 people
9am +	10 %	5 people

3) How many alarms do you set?

0	6%	3 people
1 - 2	71%	36 people
3 – 5	16%	8 people
5 +	8%	4 people

4) On a working day how much spare time do you have in the mornings?

Less than 30 minutes	57%	29 people
30 minutes – 1 hour	31%	16 people
1 hour – 2 hours	6%	3 people
More than 2 hours	6%	3 people

5) Are you often irritable in the mornings?

Yes	41%	21 people
No	27%	14 people
Sometimes	31%	16 people

6) Do you often feel demotivated throughout the day?

Yes	31%	16 people
No	27%	14 people
Sometimes	41%	21 people

7) Have you heard of a dawn light simulator and if so would you consider buying one?

I have not heard of one	49%	25 people
-------------------------	-----	-----------

Yes	25%	13 people
No	33%	17 people
I already own one	2%	1 person

8) If yes to previous question how much would you be willing to pay?

£10 - £20	50%	11 people
£20 - £30	18%	4 people
£30 - £40	32%	7 people
£40 - £50	0%	
£50 +	0%	

Figure 3: Circuit diagram

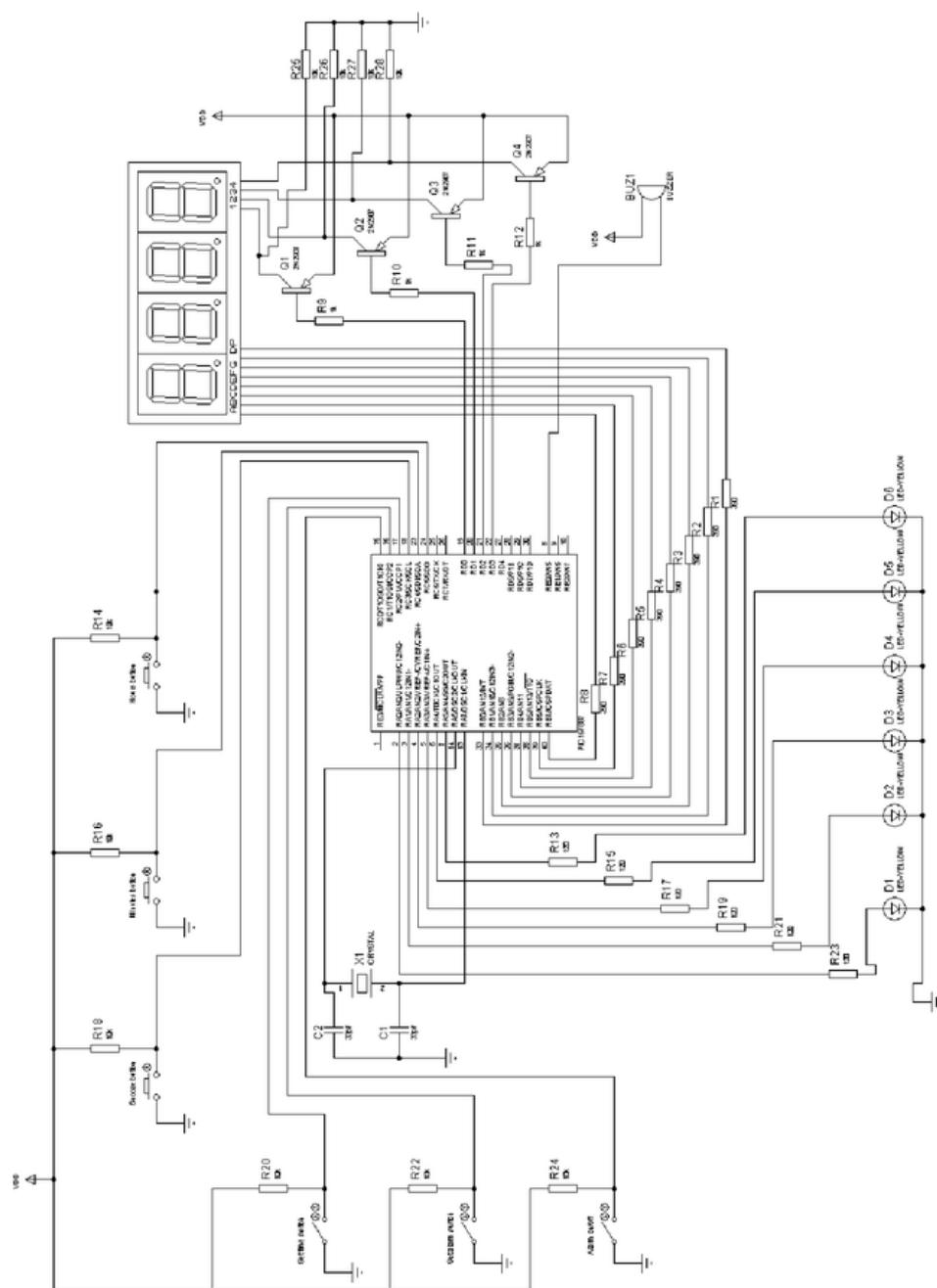


Figure 4: PIC16F887 header. File

```
#include <16F887.h>

#device ADC=10


#FUSES PUT          //Power Up Timer
#FUSES NOMCLR        //Master Clear pin used for I/O
#FUSES NOPROTECT     //Code not protected from reading
#FUSES NOCPD         //No EE protection
#FUSES BROWNOUT      //Reset when brownout detected
#FUSES IESO          //Internal External Switch Over mode enabled
#FUSES FCMEN         //Fail-safe clock monitor enabled
#FUSES NOLVP         //No low voltage prgming, B3(PIC16) or B5(PIC18) used for I/O
#FUSES BORV21        //Brownout reset at 2.1V
#FUSES NOWRT         //Program memory not write protected


#use delay(crystal=4MHz)
#use FIXED_IO( B_outputs=PIN_B7,PIN_B6,PIN_B5,PIN_B4,PIN_B3,PIN_B2,PIN_B1,PIN_B0 )

#define sev_sega      PIN_B0
#define sev_segb      PIN_B1
#define sev_segc      PIN_B2
#define sev_segd      PIN_B3
#define sev_sege      PIN_B4
#define sev_segf      PIN_B5
#define sev_segg      PIN_B6
#define sev_segh      PIN_B7
#define almo-o        PIN_C0
#define set_alar       PIN_C1
#define set_time       PIN_C2
#define snooze         PIN_C3
#define mins           PIN_C4
#define hrs            PIN_C5
```

```
#define LED PIN_D7
```

```
#define DELAY 500
```

```
#define num0 0x03
```

```
#define num1 0x9f
```

```
#define num2 0x25
```

```
#define num3 0x0d
```

```
#define num4 0x99
```

```
#define num5 0x49
```

```
#define num6 0x41
```

```
#define num7 0x1f
```

```
#define num8 0x01
```

```
#define num9 0x09
```

```
#define numa 0x11
```

```
#define numb 0xc1
```

```
#define numc 0x63
```

```
#define numd 0x85
```

```
#define nume 0x61
```

```
#define numf 0x71
```

Figure 5: Flow chart for counting time and display

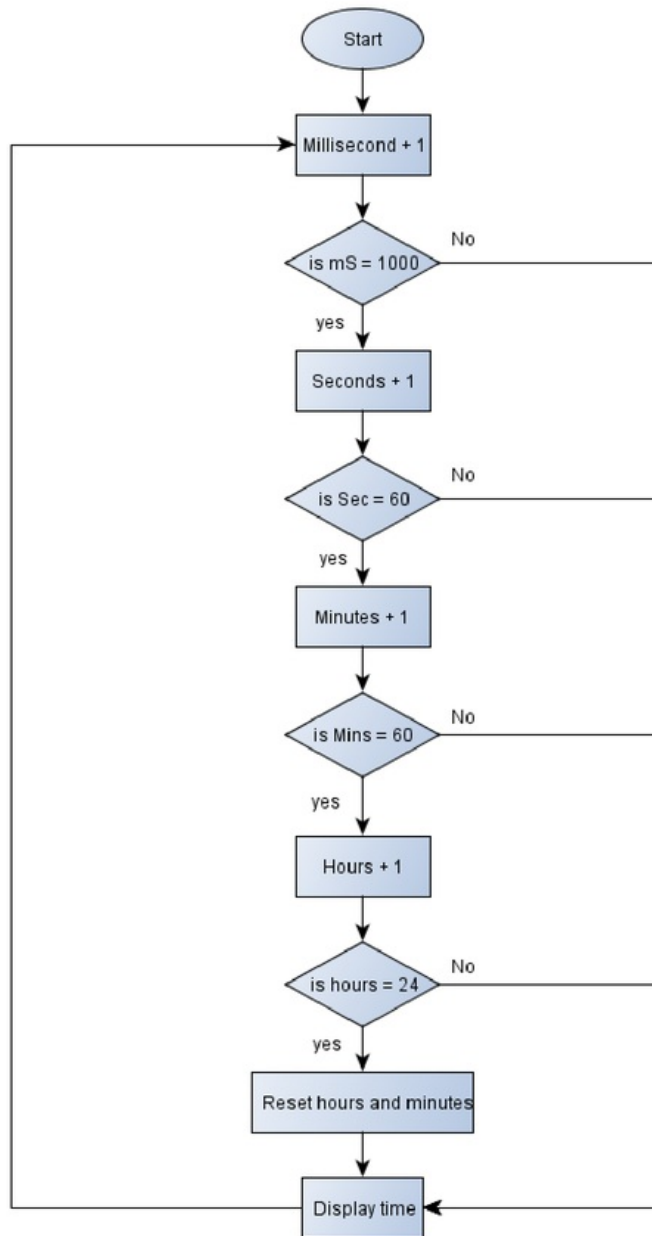


Figure 6: Flow chart for user changing time

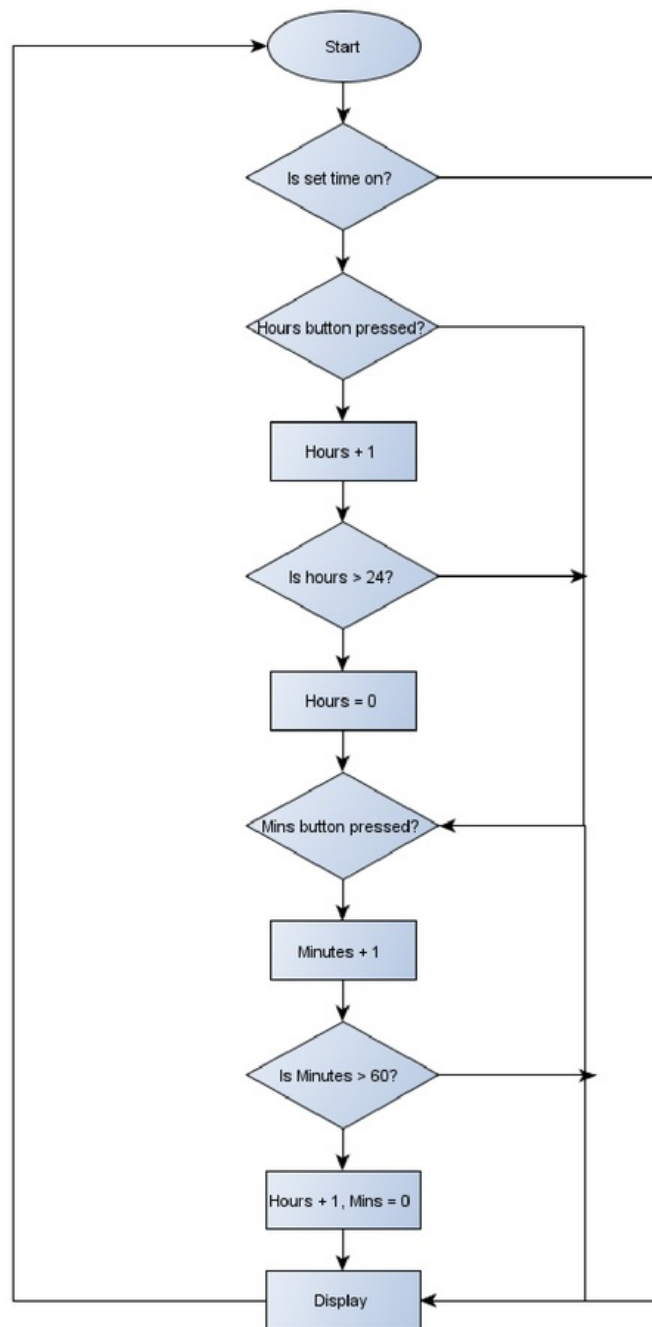


Figure 7: Flow chart for user setting alarm and sunrise alarm

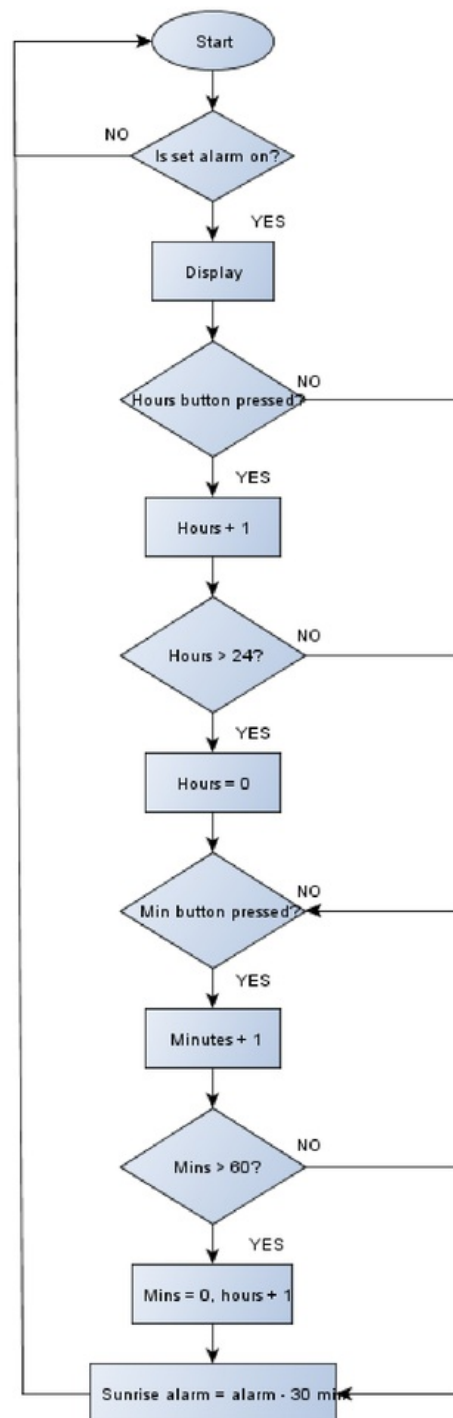


Figure 8: Flow chart for outputting alarm and sunrise simulator

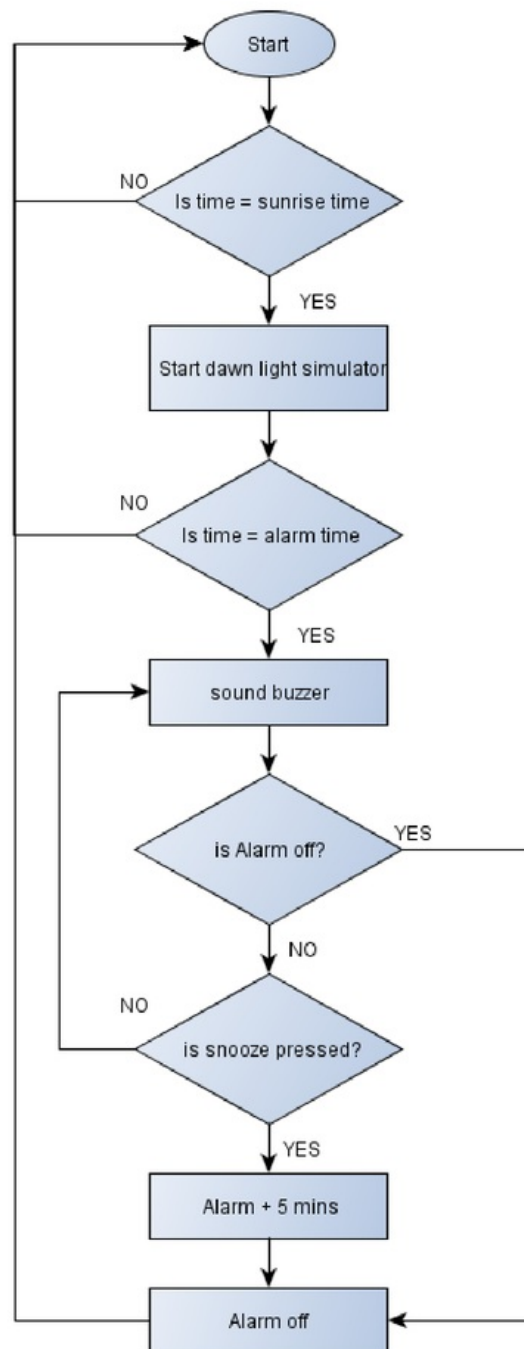


Figure 9: PIC 16f887 code

```
#include <main.h>    // includes the main.h file

#define buzzer PIN_e0 // defines Pin E0 as buzzer
#define LED1  PIN_A0 // defines pin A0 as LED1
#define LED2  PIN_A1 // defines pin A1 as LED2
#define LED3  PIN_A2 // defines pin A2 as LED3
#define LED4  PIN_A3 // defines pin A3 as LED4
#define LED5  PIN_A4 // defines pin A4 as LED5
#define LED6  PIN_A5 // defines pin A5 as LED6

unsigned long int val=0; // intreger used to store milliseconds value
unsigned int secs=0;      // intreger used to store Seconds value
unsigned int minutes=0; // intreger used to store Minutes value
unsigned int hours=0;     // intreger used to store Hours value

unsigned int alarm = 0;    // used to state if alarm is on or off
unsigned int alarm_hours=0; // intreger used to store Alarm hours for alarm
unsigned int alarm_minutes=0; // intreger used to store Alarm minutes for alarm

unsigned int sunrise_hours = 0;    // intreger used to store sunrise hours for sunrise alarm
unsigned int sunrise_minutes = 0;   // intreger used to store sunrise minutes for sunrise alarm
unsigned int sunlight_on = 0;       // used to state when the sunrise alarm is on

unsigned int light_on = 0;          // used to state when the light function is on
unsigned int LEDS = 0;              // used to state which LED will turn on in sunrise alarm
unsigned int LEDS_on = 0;           // used to state which led will turn on for each light up

#INT_TIMER2
void TIMER2_isr(void) // Timer 2 interrupt (1mS)
{
    val++; // millisecond counter
```

```

if(val==1000) //second counter
{
    secs++; // used to increment seconds
    val=0;  // reset millisecond
    if(secs==60) // minute counter
    {
        minutes++; // used to increment minutes
        secs=0;      // // resets seconds
        if(minutes==60) // hour counter
        {
            hours++; // used to increment hours
            minutes=0; // resets minutes
            if(hours==24) // day counter
            {
                hours=0; // resets hours
                minutes=0; // resets minutes
            }
        }
    }
}

}

void main()
{
    char message[5]="0123"; // Used to store an array for multiplexing
    int q; // variable for selecting 7 segment display
    int a; // variable to selecting 7 segment value

    setup_timer_2(T2_DIV_BY_4,24,10); //100 us overflow, 1.0 ms interrupt

```

```

enable_interrupts(INT_TIMER2);    // allows Timer to to interrupt code (1mS)
enable_interrupts(GLOBAL);    // allows interrupts through code

while(TRUE)    // infinite loop
{
    //TODO: User Code

    if(!input(set_alarm)) //if PIN_C1 is 0 use this statement for alarm display
    {
        sprintf(message,"%02u%02u",alarm_hours,alarm_minutes); // convert decimal to hex
for alarm
    }
    else    // other wise use this statement for time display
    {
        sprintf(message,"%02u%02u",hours,minutes);    // convert decimal to hex for
time
    }

    for (q=0;q<4;q++)    // loop for displaying LED multiplexing showing 1 7segment at a time
with a value
    {
        a=message[3-q];    // use a to store 1 part of an array to display
        delay_ms(10);    // 1 ms delay allow the LED 7 segment display to be seen (to
be displayed on protues higher value is needed)
        output_d(0xff);    // Puts all PORTB high ready for output (Multiplexing)
        switch(q)    // use q as a variable to display select 7 segment
        {
            case 0:    output_low(PIN_D3);    // used to select Pin D3 low allowing
displayed
                        break;    // 7 segment for hours (ten) to be

            case 1:    output_low(PIN_D2);    // used to select Pin D2 low allowing
displayed
                        break;    // 7 segment for hours (units) to be

```

```

case 2:      output_low(PIN_D1); // used to select Pin D1 low allowing
            break;                // 7 segment for minutes (ten) to be
displayed

case 3:      output_low(PIN_D0); // used to select Pin D0 low allowing
            break;                // 7 segment for minutes (unit) to
be displayed

default:     break;
}
output_b(0xff); // used to set PORTB high ready for outputting the 7
segment value

switch(a)
{
case '0' : output_b(num0); // output 0 on led display
            break;
case '1' : output_b(num1); // output 1 on led display
            break;
case '2' : output_b(num2); // output 2 on led display
            break;
case '3' : output_b(num3); // output 3 on led display
            break;
case '4' : output_b(num4); // output 4 on led display
            break;
case '5' : output_b(num5); // output 5 on led display
            break;
case '6' : output_b(num6); // output 6 on led display
            break;
case '7' : output_b(num7); // output 7 on led display
            break;
case '8' : output_b(num8); // output 8 on led display
            break;
case '9' : output_b(num9); // output 9 on led display
            break;
case 'A' : output_b(numa); // output A on led display
            break;

```

```

        case 'B' : output_b(numb); // output B on led display
            break;
        case 'C' : output_b(numc); // output C on led display
            break;
        case 'D' : output_b(numd); // output D on led display
            break;
        case 'E' : output_b(ume); // output E on led display
            break;
        case 'F' : output_b(numf); // output F on led display
            break;
        default : break;
    }
    if(q==2) output_low(PIN_B0); // set PIN B0 low for Decimal Place for Hours unit 7
segment
}

// set the time loop
if(!input(set_time)) // When pin C2 is low (set time switch) use this statement
{
    if(!input(hrs)) // If Pin C5 (Hours button pressed) use this staement
    {
        delay_ms(500); // Delay for debouncing
        hours++; // incrementing hours
        delay_ms(500); // Delay for debouncing
        if(hours==24) // if the value hours is 24 use this statment
        {
            hours=0; // reset hours
        }
    }
    if(!input(mins)) // If Pin C4 (Minutes button pressed) use this
staement
{

```

```

        delay_ms(500);    // Delay for debouncing
        minutes++;        // incrementing minutes
        delay_ms(500);    // Delay for debouncing
        if(minutes==60)   //if the value minutes is 60 use this statement
        {
            hours++;       // Hour increment
            minutes=0;     // resets minutes to 0
        }
    }
}

// end of setting time

//set the alarm and sunrise alarm
if(!input(set_alarm))    // When pin C1 is low (set Alarm switch) use this statement
{
    if(!input(hrs))       // If Pin C5 (Hours button pressed) use this
    {
        delay_ms(500);    // Delay for debouncing
        alarm_hours++;     // incrementing hours
        delay_ms(500);    // Delay for debouncing
        if(alarm_hours==24) // if the value hours is 24 use this statment
        {
            alarm_hours=0;    // reset hours
        }
    }
    if(!input(mins))      // If Pin C4 (Minutes button pressed) use this
    {
        delay_ms(500);    // Delay for debouncing
        alarm_minutes++;   // incrementing minutes
        delay_ms(500);    // Delay for debouncing
    }
}

```

staement

staement

```

statement      if(alarm_minutes==60)      //if the value minutes is 60 use this

{
    alarm_hours++;          // Hour increment
    alarm_minutes=0;        // resets minutes to 0
}
}

sunrise_hours = alarm_hours; // sets the sunrise hours to alarm hours
sunrise_minutes = alarm_minutes; // sets the sunrise minutes to alarm

minutes

int sunrise = 30; // sunrise is used to loop 30 minute count down

for(sunrise = 30; sunrise != 0; sunrise --)
{
    if(sunrise_minutes == 0 )      // if sunrise minutes is equal to 0
    {
        sunrise_hours = sunrise_hours - 1;          // subtract 1 from sunrise

hours

        sunrise_minutes = 59; // set sunrise minutes to 59
    }

    sunrise_minutes = sunrise_minutes -1; // subtract 1 from sunrise minutes
}

}

// end of setting alarm and sunrise alarm

if(!input(PIN_C0)) && (hours == alarm_hours) && (minutes == alarm_minutes))
// Checks if alarm is on and if alarm is equal to the time
{
    alarm = 1;    // set alarm on

```

```

}

if(alarm == 1) // If alarm is 1 use this statement
{
    output_low(buzzer); // set buzzer low to turn alarm on

}

if(input(PIN_C0) == 1) // if alarm switch is turned off use this statment
{
    alarm = 0;    // reset alarm to 0

}

if(alarm == 0) // is alarm is 0 use this statement
{
    output_high(buzzer); // Alarm is set high (off)
}

if(!input(snooze))    //if pin C3 is pressed for snooze use this statement
{
    delay_ms(2000);
    int i = 5;          // intreger i used to store value for snooze duration
    alarm = 0;          // sets alarm to 0 to turn buzzer off

    for(i = 5; i != 0; i --) // loop 5 times to add 5 minutes to the alarm
    {
        alarm_minutes = alarm_minutes + 1;    // increment alarm minutes by 1

        if(alarm_minutes == 60)    // if alarm equals 60 use this if
        {
            alarm_minutes = 0;    // resets alarm minutes
            alarm_hours = + 1;    // increments alarm hours by 1
        }
    }
}

```

```

    }

}

    if(!input(PIN_C0)) && (Sunrise_hours == hours) && (sunrise_minutes == minutes))
// is the alarm is on and the time matches the sunrise time use this statement
{

    int counter = 5;    // intreger used to store miniutes between LED's turning on

    sunlight_on = 1;    // sets sunlight_on to 1 to state the sunrise has started

    if(LEDS == 0) // if value LED is 0 use this statment
    {
        output_high(LED1);    // turn LED 1 on
    }

    if(LEDS == 1) // if value LED is 1 use this statment
    {
        output_high(LED2);    // turn LED 2 on
    }

    if(LEDS == 2) // if value LED is 2 use this statment
    {
        output_high(LED3);    // turn LED 3 on
    }

    if(LEDS == 3) // if value LED is 3 use this statment
    {
        output_high(LED4);    // turn LED 4 on
    }

    if(LEDS == 4) // if value LED is 4 use this statment

```

```

{
    output_high(LED5);    // turn LED 5 on
}

if(LEDS == 5) // if value LED is 5 use this statment
{
    output_high(LED6);    // turn LED 6 on
    LEDS = 0;             // resets LED to 0 for next sinrise alarm
}

for(counter = 5; counter != 0; counter --)    // loop used to count 5 minutes
{
    if(sunrise_minutes == 60)    // if sunrise minutes is 60 use this statement
    {
        sunrise_minutes = 0;      // resets sunrise minutes to 0
        sunrise_hours = sunrise_hours + 1;    // increments sunrise hours by 1
    }
    sunrise_minutes = sunrise_minutes + 1; // increments sunrise minutes by 1

}

LEDS = LEDS + 1;    // increments LED by 1

}

```

if((input(PIN_C0) == 1) && (sunlight_on == 1)) // when alarm is off and sunlight simulator is on use this statement

```

{
    output_low(LED1); // turn LED1 off
    output_low(LED2); // turn LED2 off
    output_low(LED3); // turn LED3 off
}

```

```

        output_low(LED4); // turn LED4 off
        output_low(LED5); // turn LED5 off
        output_low(LED6); // turn LED6 off
        sunlight_on = 0;    // turns sunlight simulator off
    }

    // light on using hours button
    if(!input(hrs) && (light_on == 0)) // when the hours button is predd and the light is
off use this statement
    {
        delay_ms(1000);    // delay for de-bouncing
        output_high(LED1); // turn LED 1 on
        output_high(LED2); // turn LED 2 on
        output_high(LED3); // turn LED 3 on
        output_high(LED4); // turn LED 4 on
        output_high(LED5); // turn LED 5 on
        output_high(LED6); // turn LED 6 on
        delay_ms(1000);    // delay for de-bouncing
        light_on = 1; // used to state the light is on

    }

    // lights off using hours button
    1)) if(!input(hrs) && (light_on == 1) && (input(set_time) == 1) && (input(set_alar) ==
// when the hours button is predd and the light is on use this statement
    {
        delay_ms(1000);    // delay for de-bouncing
        output_low(LED1); // turn LED 1 off
        output_low(LED2); // turn LED 2 off
        output_low(LED3); // turn LED 3 off
        output_low(LED4); // turn LED 4 off
        output_low(LED5); // turn LED 5 off
        output_low(LED6); // turn LED 6 off
    }

```

```

        delay_ms(1000);    // delay for de-bouncing
        light_on = 0; // used to state the light is off

    }

    // turning on the leds 1 at a time
    if(!input(mins) && (LEDS_on == 0) && (input(set_time) == 1) && (input(set_alar) ==
1)) // if the minuts button is pressed and LEDS_on is 0 use this statment
    {
        delay_ms(1000);    // delay for de-bouncing
        output_high(LED1); // turn LED 1 on
        LEDS_on = LEDS_on + 1; // increments by 1 for next press
        delay_ms(1000);    // delay for de-bouncing
    }

    if(!input(mins) && (LEDS_on == 1) && (input(set_time) == 1) && (input(set_alar) ==
1)) // if the minuts button is pressed and LEDS_on is 1 use this statment
    {
        delay_ms(1000);    // delay for de-bouncing
        output_high(LED2); // turn LED 2 on
        LEDS_on = LEDS_on + 1; // increments by 1 for next press
        delay_ms(1000);    // delay for de-bouncing
    }

    if(!input(mins) && (LEDS_on == 2) && (input(set_time) == 1) && (input(set_alar) ==
1)) // if the minuts button is pressed and LEDS_on is 2 use this statment
    {
        delay_ms(1000);    // delay for de-bouncing
        output_high(LED3); // turn LED 3 on
        LEDS_on = LEDS_on + 1; // increments by 1 for next press
        delay_ms(1000);    // delay for de-bouncing
    }

```

```

1))      if(!input(mins) && (LEDS_on == 3) && (input(set_time) == 1) && (input(set_alar) ==
// if the minuts button is pressed and LEDS_on is 3 use this statment
{
    delay_ms(1000);    // delay for de-bouncing
    output_high(LED4); // turn LED 4 on
    LEDS_on = LEDS_on + 1; // increments by 1 for next press
    delay_ms(1000);    // delay for de-bouncing
}

1))      if(!input(mins) && (LEDS_on == 4) && (input(set_time) == 1) && (input(set_alar) ==
// if the minuts button is pressed and LEDS_on is 4 use this statment
{
    delay_ms(1000);    // delay for de-bouncing
    output_high(LED5); // turn LED 5 on
    LEDS_on = LEDS_on + 1; // increments by 1 for next press
    delay_ms(1000);    // delay for de-bouncing
}

1))      if(!input(mins) && (LEDS_on == 5) && (input(set_time) == 1) && (input(set_alar) ==
// if the minuts button is pressed and LEDS_on is 5 use this statment
{
    delay_ms(1000);    // delay for de-bouncing
    output_high(LED6); // turn LED 6 on
    LEDS_on = LEDS_on + 1; // increments by 1 for next press
    delay_ms(1000);    // delay for de-bouncing
}

1))      if(!input(mins) && (LEDS_on == 6) && (input(set_time) == 1) && (input(set_alar) ==
// if the minuts button is pressed and LEDS_on is 6 use this statment
{
    delay_ms(1000);    // delay for de-bouncing
    output_low(LED1);  // turn LED 1 off
    output_low(LED2);  // turn LED 2 off
    output_low(LED3);  // turn LED 3 off

```

```
        output_low(LED4); // turn LED 4 off
        output_low(LED5); // turn LED 5 off
        output_low(LED6); // turn LED 6 off
        LEDS_on = 0;      // resets LED_on for next cycle
        delay_ms(1000);    // delay for de-bouncing
    }

}

}
```

Figure 10: Lumie body clock survey and answers

1) What time was the alarm set for?

Participant 1: "7 am"

Participant 2: "7:30 am"

Participant 3: "7 am"

Participant 4: "7 am"

Participant 5: "7 am"

2) Was the sun out when you woke up?

Participant 1: "Yes but was dark due to clouds"

Participant 2: "Yes"

Participant 3: "No"

Participant 4: "No"

Participant 5: "No"

3) Did you wake up before the alarm?

Participant 1: "No"

Participant 2: "No"

Participant 3: "Just before"

Participant 4: "No"

Participant 5: "Yes woke up with the light"

4) Did you press snooze? If yes how many times?

Participant 1: "No"

Participant 2: "No"

Participant 3: "No"

Participant 4: "No"

Participant 5: "No"

5) Do you feel refreshed waking up?

Participant 1: "Yes"

Participant 2: "Yes"

Participant 3: "Yes"

Participant 4: "No more than usual"

Participant 5: "A little"

6) Was the device simple to use?

Participant 1: "No"

Participant 2: "No"

Participant 3: "No"

Participant 4: "It was ok"

Participant 5: "No"

7) Do you feel the light simulated to sun?

Participant 1: "Yes (Possibly better due to typical weather)"

Participant 2: "Yes"

Participant 3: "Yes"

Participant 4: "Yes"

Participant 5: "No"

8) Did you feel more motivated throughout the day?

Participant 1: "Yes"

Participant 2: "Yes more than normal"

Participant 3: "No"

Participant 4: "No"

Participant 5: "Yes a little"

9) How much would you be willing to pay for this device?

Participant 1: "£35 - £40"

Participant 2: "£30 - £40"

Participant 3: "£45"

Participant 4: "£35 - £50"

Participant 5: "£30 - £40"

10) Was there anything in particular you liked about this device?

Participant 1: "The sound (calming, natural & soothing)"

Participant 2: "The colour of the light"

Participant 3: "No"

Participant 4: "No"

Participant 5: "The design"

11) Was there anything in particular you disliked about this device?

Participant 1: "The alarm sound could be a bit louder as if in a deep sleep I personally would not wake up to it. And no snooze button"

Participant 2: "When the light was on the clock was very hot to touch"

Participant 3: "No"

Participant 4: "No"

Participant 5: "The time display lit up all night. And the heat on the lampshade from the bulb"

12) Is there anything you would like to see included within this device?

Participant 1: "Snooze button, Radio"

Participant 2: "More alarm sounds"

Participant 3: "No"

Participant 4: "No"

Participant 5: "A button to display the time when needed. More choice with the alarm sound"

13) Please write a small statement about your experiences with this device.

Participant 1: "Woke up refreshed and happy (Not usually me in the morning). The alarm & dawn light simulator made me wake up straight away (usually I have 5 – 10 alarms set). I felt awake throughout the day and motivated. Like previously stated, if I was in a deep sleep there would be a possibility this would not wake me up."

Participant 2: "Very nice design and simulated to sun efficiently, a bit confusing to use. A lot of heat generated from the bulb made me feel uncomfortable using while asleep"

Participant 3: "I found this device easy to use, and I liked the fact that the light came on before the alarm went off lighting the bedroom slowly"

Participant 4: "Good alarm clock but I could have slept through alarm as it did not make me want to jump out of bed"

Participant 5: "Nice design can see how the clock would be efficient to use in winter months"

Figure 11: Prototype clock survey and answers

1) What time was the alarm set for?

Participant 1: "7:30 am"

Participant 2: "7:30 am"

Participant 3: "7 am"

Participant 4: "7 am"

Participant 5: "7 am"

2) Was the sun out when you woke up?

Participant 1: "Yes but it was dark"

Participant 2: "Yes"

Participant 3: "No"

Participant 4: "No"

Participant 5: "No"

3) Did you wake up before the alarm?

Participant 1: "Yes"

Participant 2: "Yes"

Participant 3: "No"

Participant 4: "No"

Participant 5: "No"

4) Did you press snooze? If yes how many times?

Participant 1: "Yes once"

Participant 2: "No"

Participant 3: "No"

Participant 4: "Twice"

Participant 5: "Yes twice"

5) Did you feel refreshed waking up?

Participant 1: "Yes"

Participant 2: "Yes"

Participant 3: "Yes"

Participant 4: "Yes"

Participant 5: "Yes"

6) Was the device simple to use?

Participant 1: "Yes"

Participant 2: "if it was in a case yes"

Participant 3: "Yes"

Participant 4: "Yes"

Participant 5: "Yes"

7) Do you feel the light simulated to sun?

Participant 1: "Not so much the sun but definitely help me wake up. (More like a lamp)"

Participant 2: "Brightness yes colour no"

Participant 3: "Yes"

Participant 4: "Yes"

Participant 5: "Yes"

8) Did you feel more motivated throughout the day?

Participant 1: "Yes"

Participant 2: "During the first half yes"

Participant 3: "No"

Participant 4: "No"

Participant 5: "Yes"

9) How much would you be willing to pay for this device?

Participant 1: "£30"

Participant 2: "£15 - £20"

Participant 3: "£40"

Participant 4: "£35 - £50"

Participant 5: "£30 - £40"

10) Was there anything in particular you liked about this device?

Participant 1: "The colour of the LED's, I felt it woke me up better"

Participant 2: "The light options"

Participant 3: "No"

Participant 4: "The lights were nice and the display was well illuminated"

Participant 5: "Light wasn't too bright very nice to wake up to"

11) Was there anything in particular you disliked about this device?

Participant 1: "the sound it woke me up for sure, but with a fright"

Participant 2: "Display was too bright"

Participant 3: "No"

Participant 4: "No"

Participant 5: "No"

12) Is there anything you would like to see included within this device?

Participant 1: "A radio"

Participant 2: "Alarm on indications. Alarm sounds"

Participant 3: "not really found this clock / alarm very good"

Participant 4: "No"

Participant 5: "Maybe a yellow / orange lampshade to imitate sunrise"

13) Please write a small statement about your experiences with this device.

Participant 1: "I liked the clock but woke up with a jump. If the noise / alarm sounds was changed it would be perfect. I found the sun light simulator was better as they were brighter LED's and made the whole room light up."

Participant 2: "Overall easy to use would like to see sounds added to the design and an on / off switch"

Participant 3: "These two alarm clock I found to be very good for waking you up"

Participant 4: "Easy to use, lights lit up well and not too bright, good snooze button and the display was very easy on the eye and not too bright"

Participant 5: "Easy to use liked the idea of this prototype"