

assignment 2 5007

by Sean Jones

FILE	10202_SEAN_JONES_ASSIGNMENT_2_5007_126969_1431863494.DOCX		
TIME SUBMITTED	01-MAY-2018 06:44PM (UTC+0100)	WORD COUNT	5421
SUBMISSION ID	88049727	CHARACTER COUNT	31775

Assignment 2

Sean Jones

HNC electrical and electronic
engineering

Individual project

PSDE 5007

Submission date: 01.05.2018

Summary

Overall, the project was a success as the main goal set out to achieve was to investigate whether this part would have the O-rings sensed and if it would be possible to make a fully automated machine based on this concept.

Improvements could have been made on the mechanical design of the chute and test area to prevent jams and parts being mis-positioned in the test area causing them to fail. It might have been better to get a mechanical engineer to design the mechanical layout of the machine with more experience in this field of work.

The camera worked really well and programing it proved very simple. When parts where put in the same place every time it could easily identify the difference between a passed part and a fail part. The problem with the part testing was the part not consistently being positioned in the same please every time in the test area.

Contents

1. Introduction & Background Research.....	4
Number of bits	12
Memory.....	12
Instruction set	12
Memory Architecture	13
Microcontroller Ports.....	14
Microcontroller timing.....	15
2. Specification.....	22
3. Planning	23
4. Testing Plan.....	25
5. Final Solution	28
7. Discussion and Conclusions	30
Reference List and Bibliography	32
Bibliography	32

TABLE OF FIGURES

Figure 1 the use of the yoke	5
Figure 2 drawing of the yoke	5
Figure 3 - PIC Microcontroller architecture (Electronicshub, 2016).....	11
Figure 6 - Harvard Architecture	13
Figure 7 - Von-Neumann architecture	13
Figure 8 - Microcontroller Ports (MikroElektronika, 2016)	14
Figure 9 - PORT and TRIS registers (MikroElektronika, 2016)	14
Figure 10 - Crystal Oscillator	15
Figure 3 machine layout.....	28
Figure 4 actuator wiring.....	29

1. Introduction & Background Research

The problem

A customer has a problem to check that a part has been correctly assembled on a production line and seeks to modify their production to accommodate a parts checking system.

The part is an aluminium yoke with two O-rings which get placed on the part on the production line by an operator. A means of testing the correct seating of the O-ring will need to be developed

This will be an R&D prototype setup to prove the feasibility for a full automated system

This checking is to prevent bad parts going to the customer

This will be done using the resources at EBS automation.

The solution

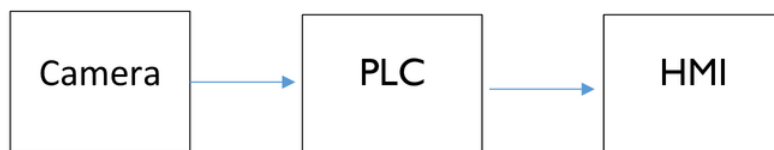
An investigation will be carried out to implement a method of testing the O-ring installation on the aluminium yoke.

Some of the alternatives that will be investigated sensors that can be used or possible camera development to test it graphically

The project will be developed during the work hours within a scheduled timing plan.

The project management will be achieved by creating a Gant chart which will indicate all processes of the project against a time line. This is to ensure targets are met and also indicate where problems and waiting time have occurred.

Two other methods of implementing this project will be investigated and one will be tested for suitability.



Project must be completed by the 25th of April 2018

The budget is £2000

The project is to find out whether a fully automated machine can be made to test a part. Due to the sensitive nature with visual sensing on a part a fully automated machine can't be built straight away as it would not be guaranteed and could lose the company a lot of money. For this project to be a success then the machine will need to consistently determine if parts are passed or failed. The slightly automated parts of the machine are only there to speed up the testing procedure of this machine.

The company has the capability of producing this machine and everything will be done in house. The part is from an external company so they have sent over information of the yoke as below in Figure 1 the use of the yoke and **Error! Reference source not found..**

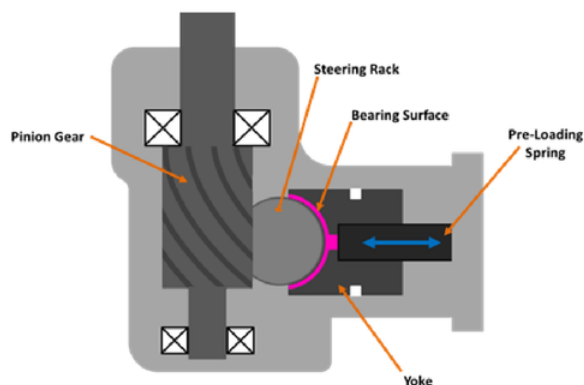


Figure 1 the use of the yoke

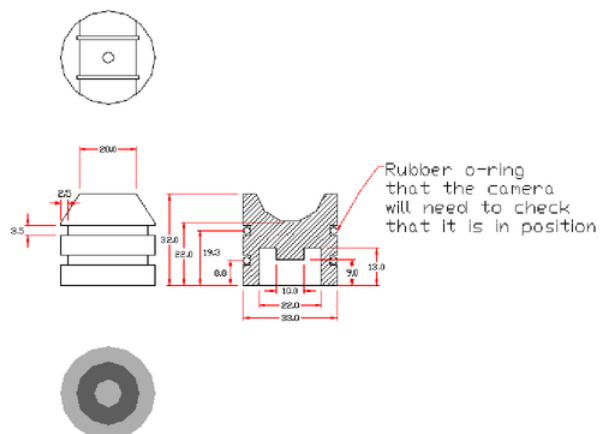


Figure 2 drawing of the yoke

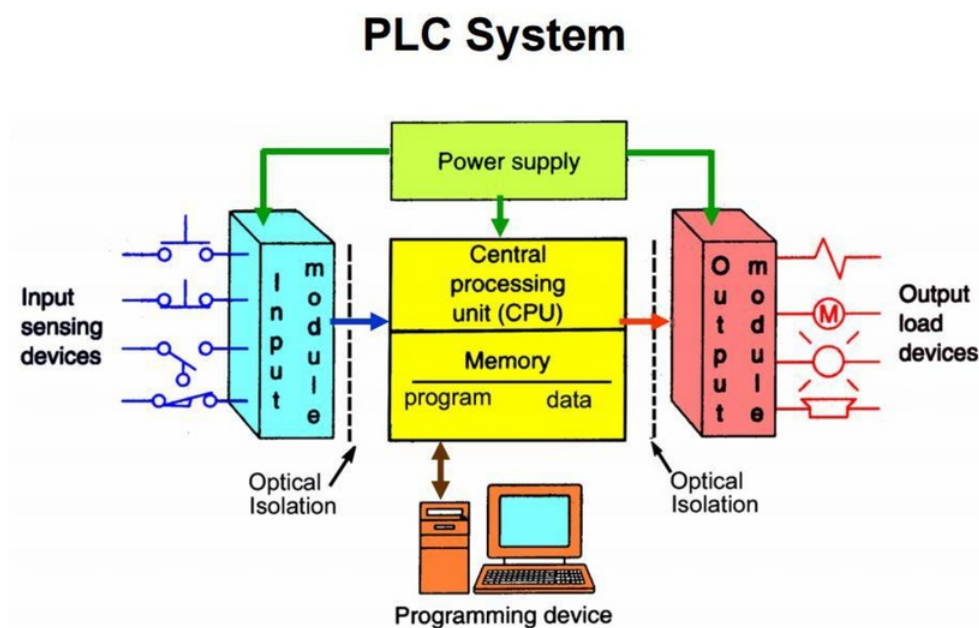
These drawings will help give a better understanding of the part. The drawing with the dimensions of the part was used to create the mechanical drawings of the machine.

A camera will need to be used to reliably sense for the O-rings. The only part of the project that has multiple options is the control of the system. These options are to use a PLC to control the system, a pic or relays and arranged hard wiring.

Using a PLC

A PLC is a very high cost approach to the project. According to (parmley-graham, 2017) the cheapest 1200 series PLC is £171.00. Additional to this high cost there is a lot of time related to the PLC route of this project. Training would be required to program the PLC and additional time scaling would be required for this training and for writing the PLC program.

Below shows the basic set up of a PLC.



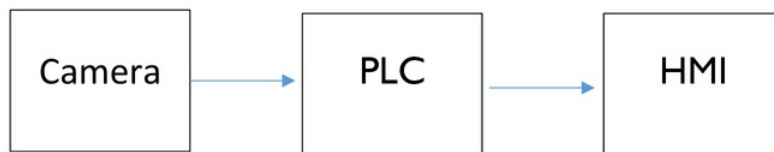
(machinedesign, 2018)

(tech-zephyr, 2018) Says "A PLC (i.e. Programmable Logic Controller) is a device that was invented to replace the necessary sequential relay circuits for machine control. The PLC works by looking at its inputs and depending upon their state, turning on/off its outputs. The user enters a program, usually via software, that gives the desired results."

The advantage over using the PLC would be the range of functions that are available. For example sequencing can be made very easy in comparison to relays as it can all be done in the logic. The time to design the electrical circuit would be reduced as there isn't any tricky relay logic which needs to be included. The greatest advantage of the PLC is future expandability which relays can't offer.

(metroid, 2018) States "A small PLC can be roughly the size of a house brick. On the other hand, a row of 10-20 relays can be twice the length, and require twice the amount of wiring

(i.e. larger cable ducts)." This will mean a less time spent on wiring the jig and a smaller enclosure used for the electrics.



A HMI can be interfaced with the PLC to give the operator full control of the system including the current image on the camera and full manual movements. A HMI can make fault finding a lot simpler because all the inputs and output states can be put on one screen for an engineer to see.

A PLC can also be useful for batch recording, for example if there was a 1000 parts running through the machine then where as you would usually have to fill in a spreadsheet to evaluate the pass and fails, the could log all the fails and list them on the HMI with a time and part number of that batch. This can help trends in failures and make the machine a lot more user friendly.

The PLC, HMI and the camera will all communicate using Ethernet. This will give fast communication and a faster running machine.

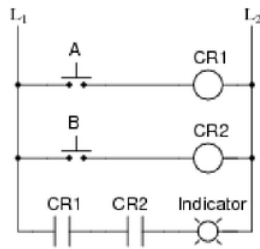


(rohtekautomation, 2018)

Relays

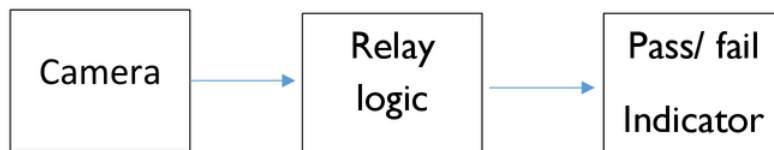
Relays are a very cheap and simple way of performing some sort of sequence. This method is only viable with small sequences as too complicated of a sequence will need too many relays and make it cost more than a PLC. (metroid, 2018) States "For small and simple control systems, relay logic can be ideal – it's low cost and easy to service."

(allaboutcircuits, 2018) "In ladder logic symbolism, an electromechanical relay coil is shown as a circle, and the contact(s) actuated by the coil as two parallel lines, almost like a capacitor symbol. Given this knowledge, interpret the following ladder logic diagram":



(allaboutcircuits, 2018)

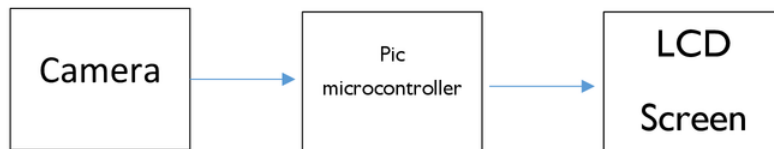
The great disadvantage of the relays are that at the very start of the project it's got to be certain that all the desired tasks that are to be performed can be done with only relays. If it is found out later that there's something that can't be done with relays then it can be very costly as a PLC may need to be bought as well and could put the project over budget.



Pic microcontroller

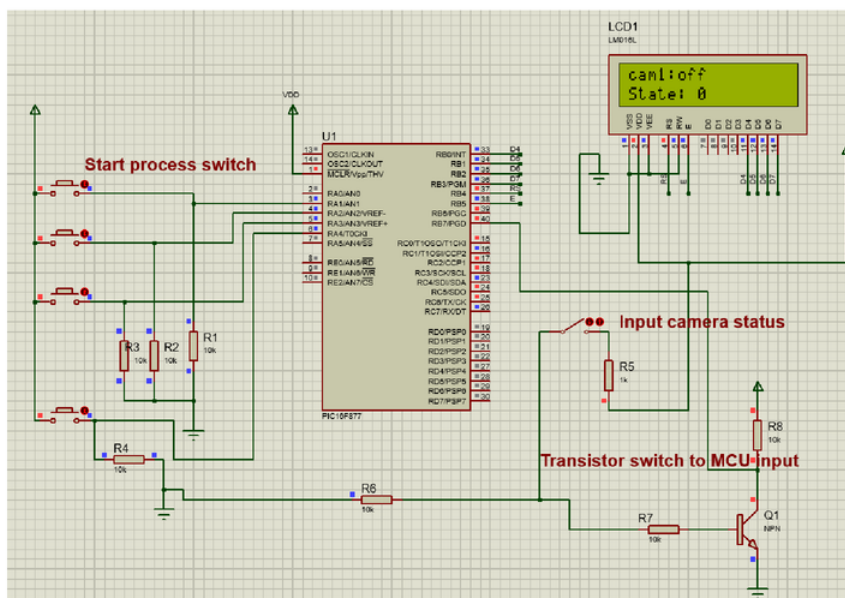
A PIC is a very small device which can perform a lot of functions. This device is also very cheap and would make the project well under budget. The Pic can also perform many functions that relays can't.

A lot of research would be required to use the PIC as EBS doesn't do anything with electronic circuits. This would create a large setback in the project which would jeopardise the plan.



Camera image control using Proteus CAD system

Proteus circuit for capture control from camera



The above circuit demonstrates how an image camera can be controlled using a MCU. The circuit is easily expandable to use more than one camera using a state process to switch the cameras.

Test results from circuit.

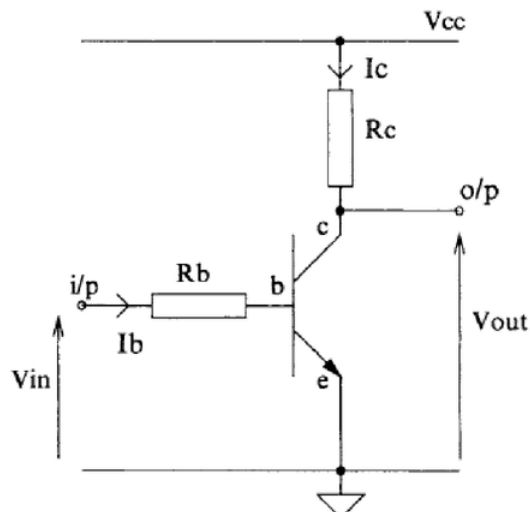
Test No	Camera status	LCD display
1	off	cam1: off State: 0

2	on	cam1: on State: 0
3	on	Cam1: complete State: 5
4	off	cam1: off State: 0

The test results show that the camera is in the off position at test number 1. Test number 2 shows the camera on and the state is in the initial position 0. Test number 3 shows the process after all 5 states and the test is complete. Test number 4 is when the process is back at the start position.

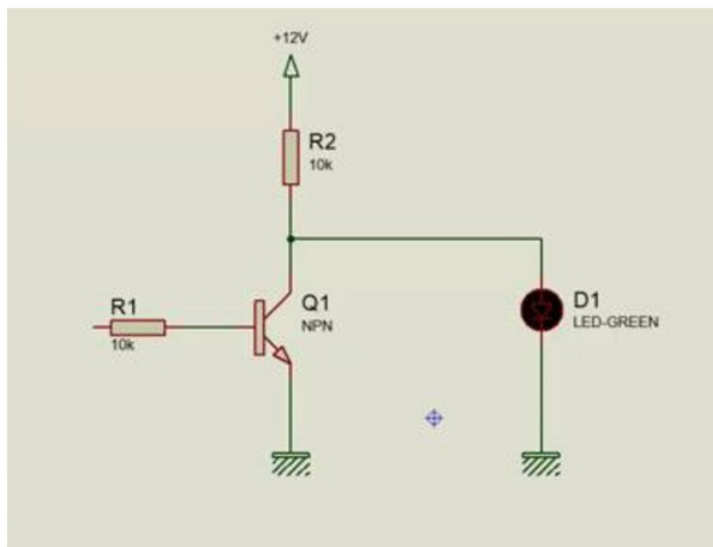
The state process can be used to add more cameras to the process and can easily be extended by manipulating the C program.

Transistor switch circuit from the camera circuit.



The transistor is like an electronic switch. It can turn a current on and off. A simple way to think of it is to look at the transistor as a relay without any moving parts. A transistor is similar to a relay in the sense that you can use it to turn something ON and OFF (P. Horowitz, W. Hill, 2015).

Circuit test using Proteus



Led will illuminate when a voltage is applied to the input base of the transistor. This circuit can now be used as an input to the MCU.

PIC Microcontroller

The PIC (Programmable Interface Controller), used in this assignment for the design of an embedded system, is a RISC architecture microcontroller available in 8, 16 and 32 bit. It has around 40 instructions that can take up to 4 clock cycles to execute. As most microcontrollers has programmable input/output ports, an embedded memory, serial and parallel ports, timers, an internal RC oscillator, ADCs and DACs. In Figure 3 below is shown the architecture of a PIC microcontroller (MicrocontrollerBoard.com, 2016).

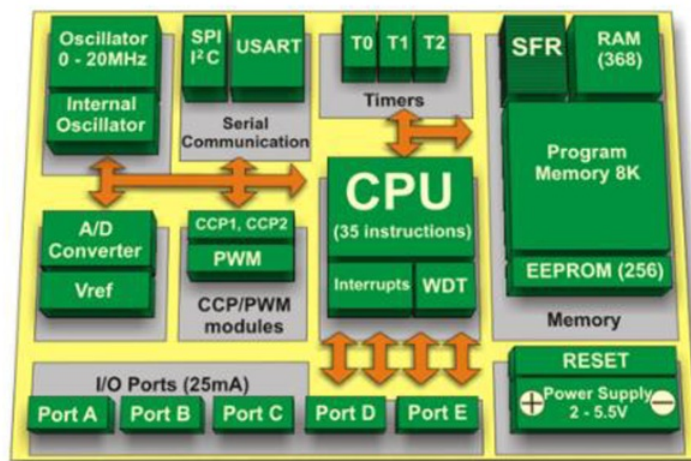


Figure 3 - PIC Microcontroller architecture (Electronicshub, 2016)

Number of bits

The number of bits define the precision and performance of the microcontroller when carrying out arithmetic and logic operations. Microcontrollers can have 8-bit, 16-bit and 32-bit microcontrollers.

Memory

The memory of microcontrollers, which allow storing information such as the code and functional blocks, can be embedded, which means that all the functional blocks are on the chip or can be external when all the functional blocks of the embedded system are not available on the microcontroller chip.

Instruction set

There are two instruction sets that can be used: CISC and RISC, which architectures are shown in **Error! Reference source not found.** and **Error! Reference source not found.** below. The first one, which stands for Complex Instruction Set Computer, allows the programmer to use an instruction that replaces many simpler instructions and therefore causes the execution of each instruction to be reduced. The second, which stands for Reduced Instruction Set Computer, allows program and data to be accessed simultaneously and instructions to operate on different registers and addressing modes causing the clock cycle per instruction to be reduced and therefore also the execution time of the instruction.

Memory Architecture

The memory architecture of a microcontroller represents the method used to exchange data between the CPU and the memory. There are two design models for the architecture of microcontrollers: the Harvard memory architecture, shown in Figure 4 below, and the Von-Neumann memory architecture, also known as Princeton architecture, shown in Figure 5 below.

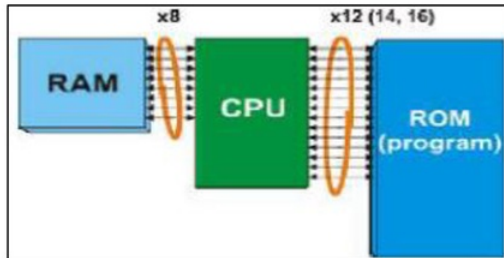


Figure 4 - Harvard Architecture

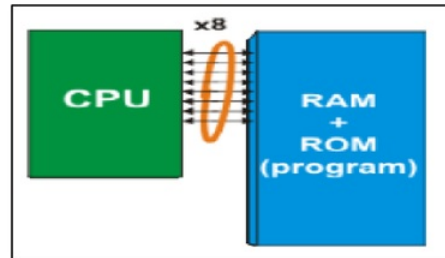
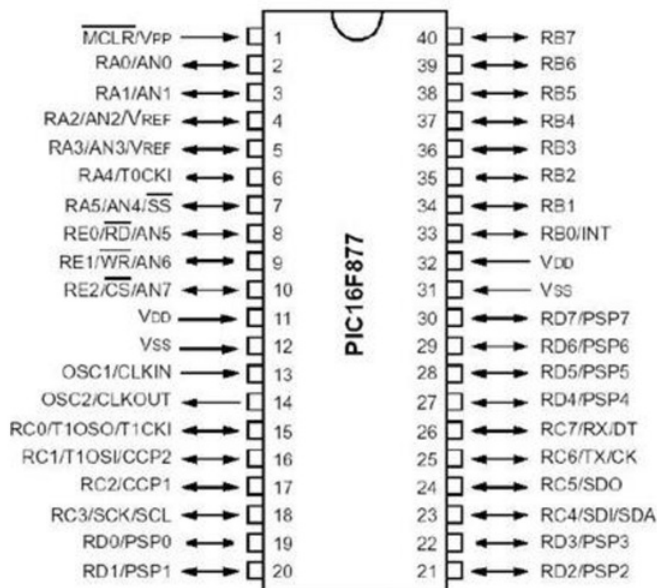


Figure 5 - Von-Neumann architecture

(Al-Aubidy, 2016)

As it can be seen from the design models shown in Figure 4 and Figure 5, in the Harvard memory architecture the memory address space for the program and the data memory are different, since the program memory is stored in the ROM while the data memory is within the RAM. For the Von-Neumann memory architecture, instead, there is a common memory address for both program, ROM, and data memory RAM (Agarwal, 2013).

The PIC controller used for this system is the PIC16F877:



PIC16F877 pin out (Morlote, 2013)

Microcontroller Ports

In a PIC microcontroller, all input and output pins used for connecting external devices, are grouped into bi-directional ports, known as port A, which is 6-bit wide, B, C and D, which are 8-bit wide. PORTE is not implemented on the PIC used for this assignment but it has three pins that can be manually configured as inputs or outputs. These ports can be configured as inputs, outputs or can be used for other functions.

Each port can be set as input or output by using the TRIS register command. If the bit of the TRIS register is 0 then the port is configured as an output and if the bit is set to 1 then the port behaves as input, as shown in Figure 6 below.

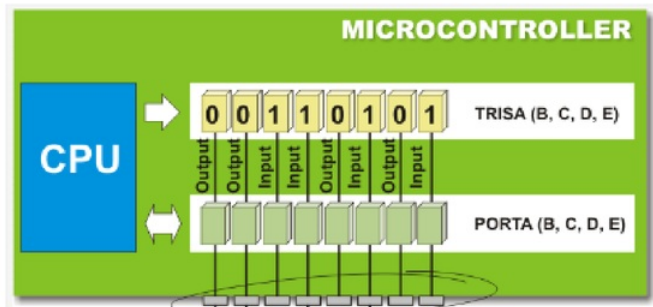


Figure 6 - Microcontroller Ports (MikroElektronika, 2016)

In order to set the bit of an output ports or read the bit of an input port is used the PORT register. As it can be seen in Figure 7 below both PORT and TRIS command are used to set the all the pins on that particular register, while for setting a specific pin i.e. of the PORTA register the RAX command can be used (MikroElektronika, 2016).

	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	R/W (x)	Features
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TRISB	TRISB7	TRISB6	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Figure 7 - PORT and TRIS registers (MikroElektronika, 2016)

Microcontroller timing

The timing of the microcontroller, or in other words how fast the program runs in scan times, depends upon the frequency of the crystal oscillator clock, which as it can be seen in Figure 8 below, can vary between 1 and 20 MHz. This can then be used as time base for a specific application.

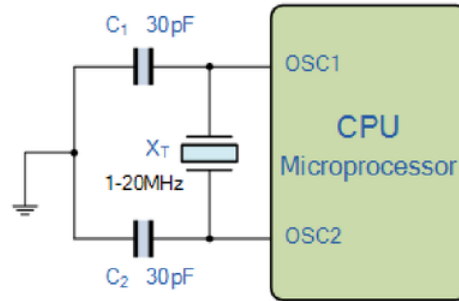


Figure 8 - Crystal Oscillator

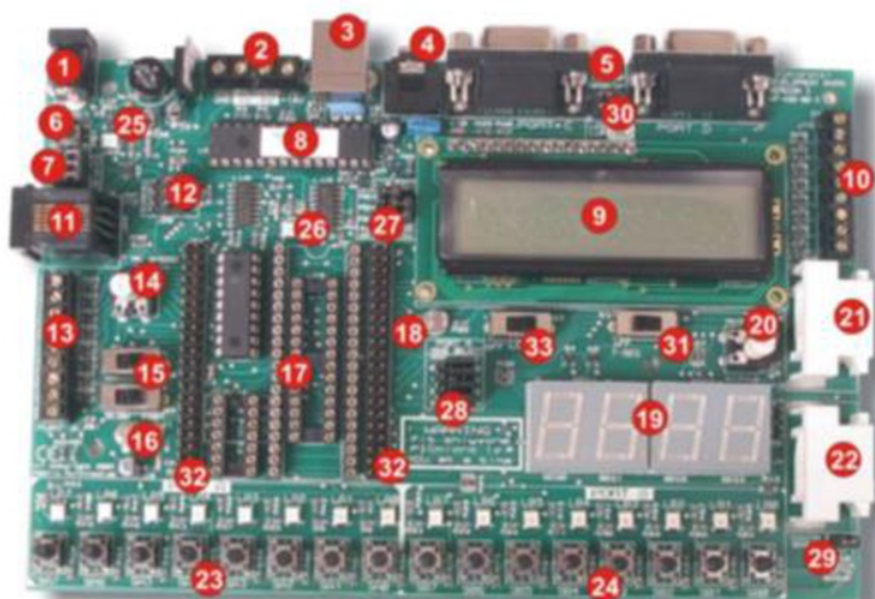
The micro system consists of 5 inputs which must be set in order for the output to be switched on. The system operates using a state machine principle whereby the next state cannot be implemented until the conditions for the previous state has been achieved.

The program was written using C code in MPLAB and HiTeCH C compiler environment. The registers for the PIC to operate with the required IO were set and tested. These are embedded into the main C program.

Table for IO allocation.

PORT A	Set for 5 inputs and 3 outputs of which only 1 is needed	Used for inputs and 1 output FOR SWITCH
PORT B	Set for all outputs.	Used for output for LCD

This system architecture which is designed and implemented using Proteus is a copy of the HP488 development board. This means that the same system will work when connected practically to the development board.



1. Power supply connector
2. 5V outputs for E-blocks connection
3. USB connector
4. Single bit audio output
5. E-blocks connectors for ports C and D
6. Reset switch
7. Power selector link block—USB or Power connector
8. USB interface chip
9. 2 line 16 character alphanumeric display
10. Screw terminal connectors for port B
11. In Circuit Debug socket
12. ICD/USB programming selection
13. Screw terminal connectors for port A
14. RC speed potentiometer
15. Oscillator mode selection switches
16. Removable crystal
17. Sockets for programming 8, 14, 18, 28 and 40 pin PICmicro devices
18. On-board light sensor
19. Quad 7-segment displays
20. Analogue input potentiometer
21. External analogue sensor input
22. External digital sensor input
23. Port A switches and LEDs
24. Port B switches and LEDs
25. Power indicator
26. Programming indicator
27. LVP pin selector
28. Port A analogue/ digital selector

29. Digital sensor selector
30. LCD display LVP mode selector
31. 7-segment display enable
32. Expansion bus
33. LCD enable

C program listing

```
#include <pic.h>
#include "LCD.h"

void DisplayPump(int line);
//Function prototypes

void DisplayState(int line);
void DisplayComplete(int line);
void Delay(int t);
//Function prototype

int state=0, i, j, pump;
//global variables

////////////////////////////////////
////////////////////////////////////
void main(void)
Function
{
    TRISA = 0xFF;
    port A = 1111 1111
    TRISB = 0x80;
    port B = 1000 0000
    TRISC = 0x00;
    port C = 0000 0000
    ADCON1= 0x07;
    ADCON1 = 0000 0111

    LCD_Startup();
    header, runs function in LCD.c

    while(1)
    super-loop
    {
        if (RB7==1) pump = 0;
        else pump = 1;

        DisplayState(1);
        on line 1 of LCD
        //DisplayPump(0);

        switch(state)
        flow depending on state value
        {
            case 0:
                DisplayPump(0);
                if(RA1) state = 1;
                equal to 1 set state to equal 1
                if(RA2) state = 5;
                break;
                //This breaks out from each case
        }
    }
}
```

```

        case 1:
            //Case 1, run if state equals 1
            DisplayPump(0);

                Delay(1000);
                state = 2;
                break;
            //This breaks out from each case

        case 2:
            //Case 2, run if state equals 2
            DisplayPump(0);

                Delay(1000);
                state = 3;
                break;
            //This breaks out from each case

        case 3:
            DisplayPump(0);
            if state equals 3
                Delay(1000);
                state = 4;
                break;
            //This breaks out from each case
            //Case 3, run

        case 4:
            //Case 4, run if state equals 4
            DisplayPump(0);

                Delay(1000);
                state = 5;
                break;
            //This breaks out from each case

        case 5:
            //Case 5, run if state equals 5
            DisplayComplete(0);
            if(RA1) state = 0;
            // Delay(1000);

            break;
            //This breaks out from each case

        case 6:
            //Case 6, run if state equals 6

                Delay(1000);
                state = 7;
                break;
            //This breaks out from each case

```

```

        case 7:

            //Case 7, run if state equals 7
            Delay(1000);
            state = 8;
            break;
        //This breaks out from each case

        case 8:
            //Case 8, run if state equals 8

            Delay(1000);
            state = 9;
            break;
        //This breaks out from each case

        case 9:
            //Case 9, run if state equals 9

            Delay(1000);
            state = 0;
            break;
        //This breaks out from each case

    }
    while(RA1 || RA2 || RA3);                                //While RA1
    or RA2 or RA3 are active do nothing This freezes the display form setting the variables
    }
}

////////////////////////////////////////Function prototype for
delay////////////////////////////////////////

void Delay(int t)
{
    for(i=0;i<t;i++) // Outer loop
    {
        for(j=0;j<50;j++); // inner loop
    }
}

////////////////////////////////////////Funtion prototype to display the
state////////////////////////////////////////

void DisplayState(int line)
{
    char state_string[10] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
    state_string[0] = 'S';

```

```

state_string[1] = 't';
state_string[2] = 'a';
state_string[3] = 't';
state_string[4] = 'e';
state_string[5] = ':';
state_string[6] = ' ';
state_string[7] = (char) state + 48;
state_string[8] = 0;

    DisplayString(line, state_string);
}

void DisplayPump(int line)
{

char pump_string[14] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

if (pump == 1)
{
pump_string[0] = 'c';
pump_string[1] = 'a';
pump_string[2] = 'm';
pump_string[3] = '1';
pump_string[4] = ':';
pump_string[5] = 'o';
pump_string[6] = 'n';
pump_string[7] = ' ';
pump_string[8] = ' ';
pump_string[9] = ' ';
pump_string[10] = ' ';
pump_string[11] = ' ';
pump_string[12] = ' ';
pump_string[13] = 0;

}

if (pump == 0)
{
pump_string[0] = 'c';
pump_string[1] = 'a';
pump_string[2] = 'm';
pump_string[3] = '1';
pump_string[4] = ':';
pump_string[5] = 'o';
pump_string[6] = 'f';
pump_string[7] = 'f';
pump_string[8] = ' ';
pump_string[9] = ' ';
pump_string[10] = ' ';
pump_string[11] = ' ';
pump_string[12] = ' ';
pump_string[13] = 0;

```

```

}
DisplayString(line, pump_string);
}

void DisplayComplete(int line)
{
    char state_string[14] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
    state_string[0] = 'C';
    state_string[1] = 'a';
    state_string[2] = 'p';
    state_string[3] = 't';
    state_string[4] = ':';
    state_string[5] = 'c';
    state_string[6] = 'o';
    state_string[7] = 'm';
    state_string[8] = 'p';
    state_string[9] = 'l';
    state_string[10] = 'e';
    state_string[11] = 't';
    state_string[12] = 'e';
    state_string[13] = 0;

    DisplayString(line, state_string);
}

```

2. Specification

For the system to pass final testing it must meet all the following information.

Mechanical features

- Be no greater than 500mm³ excluding the electrical enclosure
- Have guarding in high risk areas.
- Parts must not be dented or scratched throughout the process.

Electrical features

- A IFM o2d222 camera must be used to sense an O-ring.
- All 24vdc cables must be of thickness of 0.75mm².
- The circuit must be easy to modify as this is a development project.

Additional information

- The machine will have a start button which will be green. The purpose of this button is to start the sequence.
- There will be a reset switch which will reset the machine if there is a failed part.
- The machine will have a red light to say when a part has failed this light will remain on until a reset switch is turned.
- Have a chute so parts can be back filled
- The machine must be semi-automated and have the operator must only have to inset parts into the chute, collect them from the passed area and remove them from the test area when failed.
- The machine must move the part from the test area into the pass area when passed.
- The cycle time must be at least 1 part every 10 seconds.
- Failed parts must be removed manually.
- A part will be deemed as a pass if both O-rings are in place.
- A part will be deemed a fail if either or neither of the O-ring are not in place.

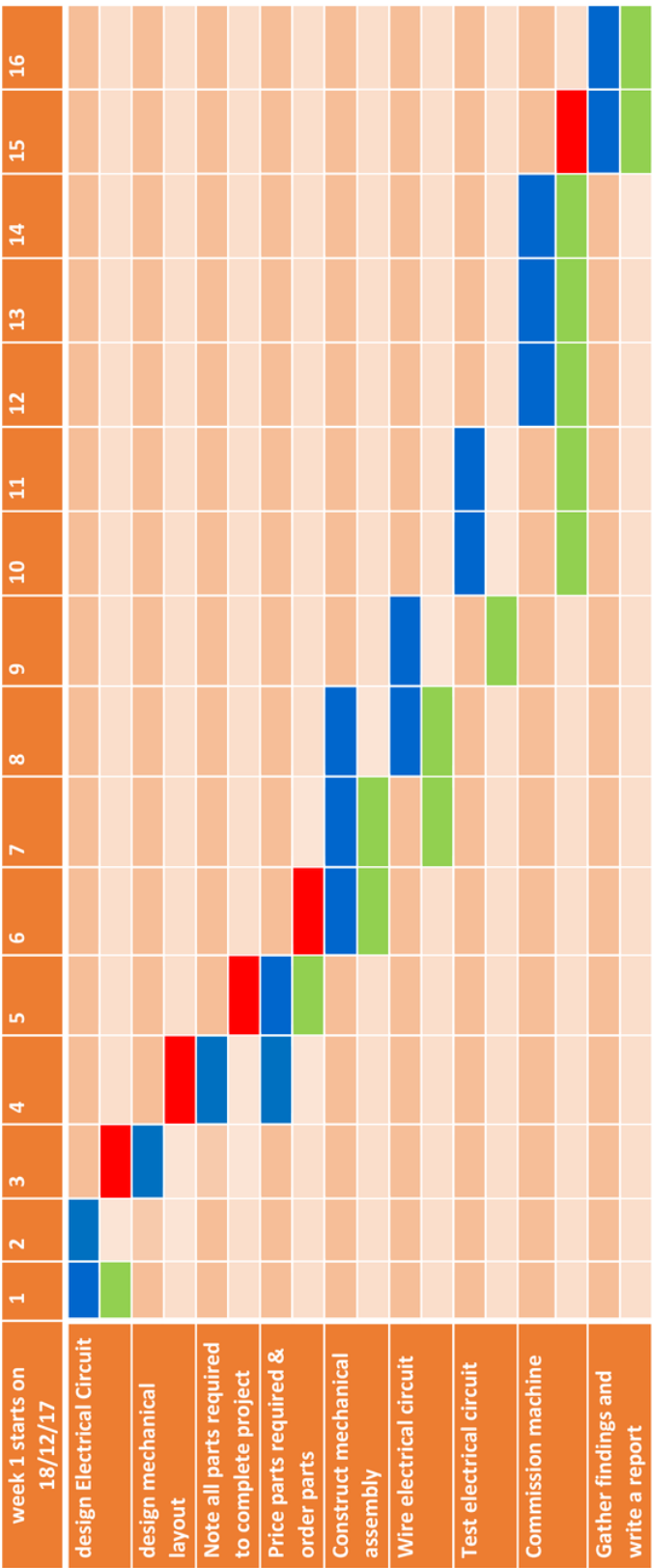
The budget of £2000 should be strictly followed and if the budget looks as if it is going to get overshoot then a meeting with the supervisor must take place to take action to either increase the budget or find a solution around the problem. The budget is only for the bought out parts of the machine. Software, training or hourly production rate is not included in this budget.

3. Planning

The project will last over 16 weeks starting from the 18th of December 2017. To keep track of the project and to ensure it doesn't run on for too long there will be a gant chart used to monitor the progress. A gant chart is the best way to represent this as it shows a graphical representation of the milestones being met. If a milestone is not met then it will be marked on the gant chart to show this. Greater work input will have to be put in if a milestone is not met to ensure that the project doesn't fall behind schedule.

The 4th week is the current week of the plan. The project is currently 1 week behind schedule. This was due to the oversight that a week will be lost for the Christmas holiday which put the project behind schedule. Assist may be needed on the mechanical assembly to get it done quicker to get the project back on track and prevent the project being late

GANTT CHART SHOWING THE PLAN OF THE PROJECT



Good Gant chart

4. Testing Plan

The system will have a critically test to ensure the system meets the specification.

The system will be broken down into work areas where individual components can be bench tested before implementation.

Tests will be carried out at work site to satisfy that the original specification is met.

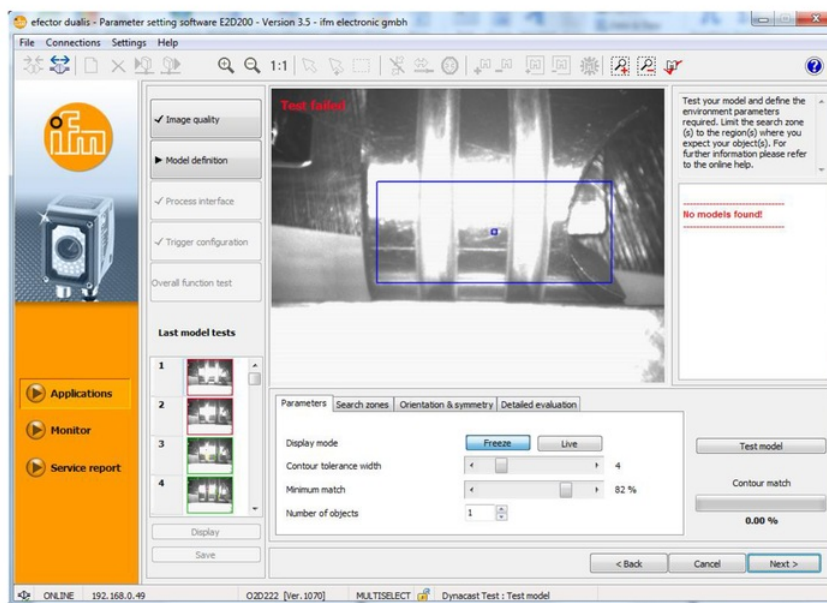
Any alterations will be discussed by management and engineers where compromises may be necessary.

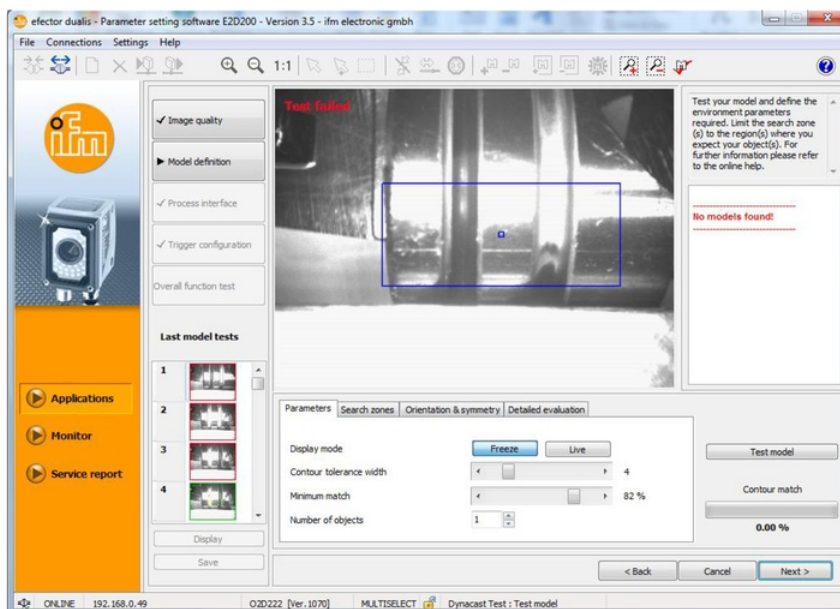
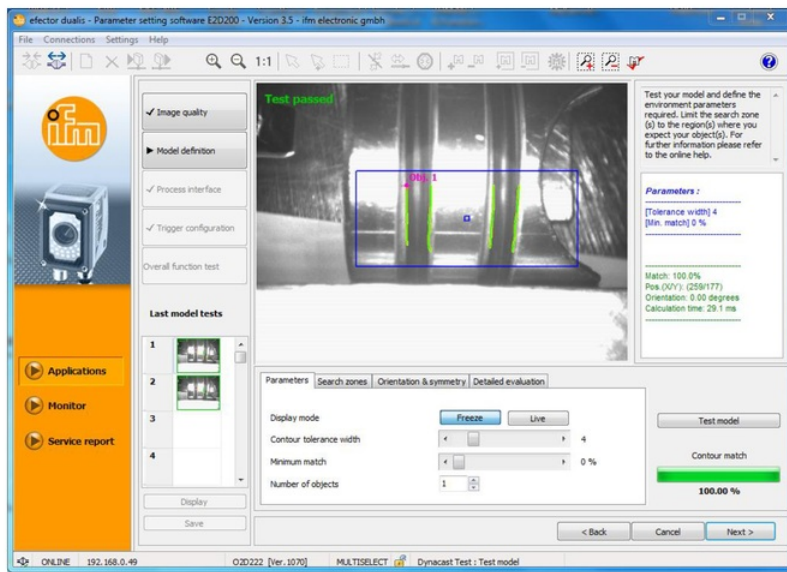
A table of results will be produced which will show how the end product performs against the specifications.

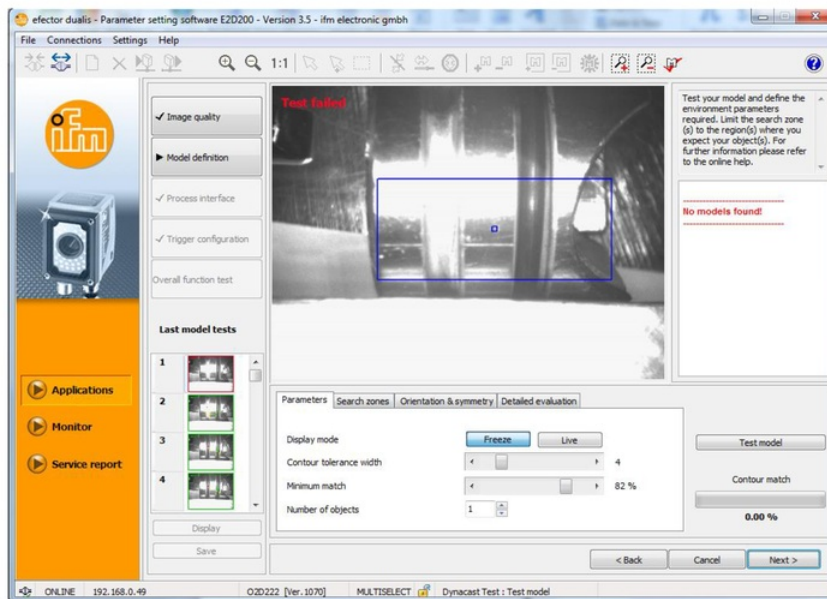
The machine must be proven to repeatable so it must continuously run to make sure the machine is consistent.

The testing showed the sequence was inconsistent and a testing table was hard to derive due to a pass was very unlikely. Although the Camera test images showed constancy when the parts were placed in the same area every time.

Below are the pass and fail photos of the camera.







The images show how the camera easily identifies the part as a pass or fail even if the part is not placed in the exact same place every time.

More information on the test is needed. How was the camera implemented with the PLC. The PLC program should be evaluated.

5. Final Solution

It was chosen to go with the relays to control the system as it would easily be designed and wired also it was chosen due to plenty of previous experience with relays so design time and commissioning time would be reduced. The cost of the relays will help the project be within budget easily. About 4 relays will be needed to perform all the functions required.

The image below Figure 3 shows what the rough layout of the system may look like. This shows the potential layout of the camera, chute and passed part actuator. The camera has the same position in relation to the part that was tested previously. This is to prevent any unwanted delays in the project where the camera may not work as it did when being tested.

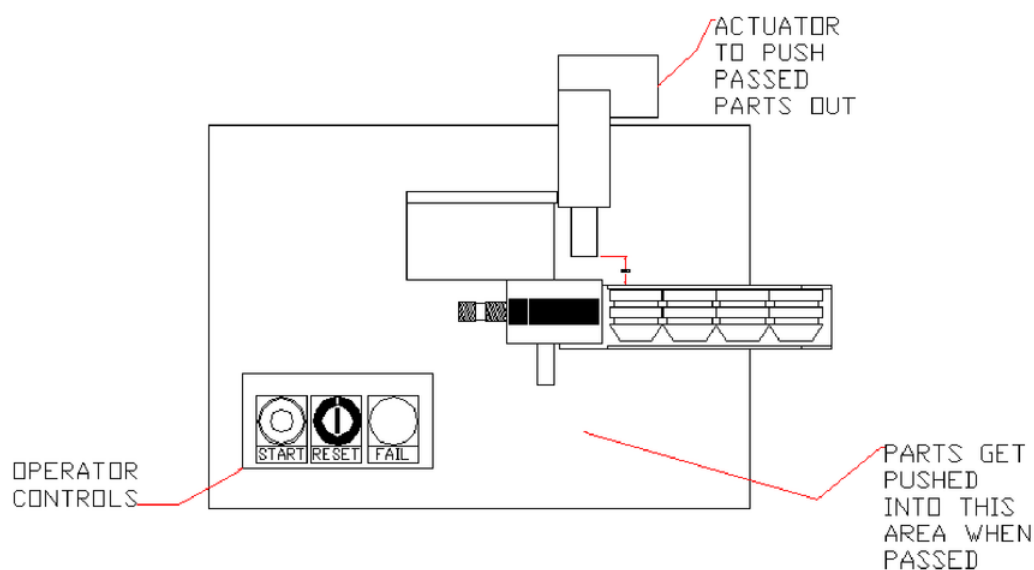


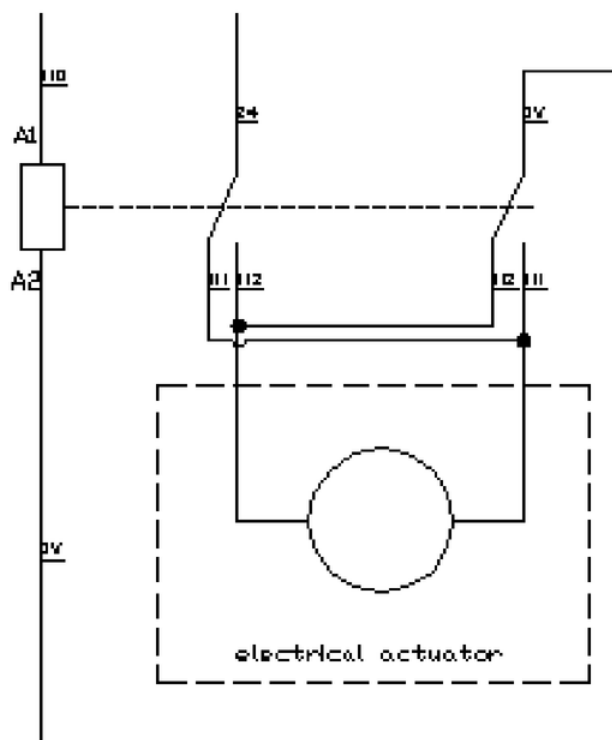
Figure 9 machine layout

The buttons are kept away from the main part of the system so an operator won't have their fingers near the moving parts of the machine when the machine starts. It's also placed over to the left so the operator can press start easily with the opposite hand that filled the chute up.

There is a part present sensor at the bottom of the chute that will sense a part to trigger the camera. This sensor would have to be set up so that it can't see the arm of the actuator when it pushes a passed part out.

The electrical diagram was drawn up. This is issue 1 for the electrical drawings as later modifications may be needed when the system is being commissioned. This is normal where overseen problems may affect the electrical circuit design.

The electrical actuator works off to wires where the polarity of the current placed on the device will determine which direction of which the actuator will move in. this can be done easily in an electrical circuit with 1 2-pole relay, as shown in Figure 4.



The camera used was an IFM O2d222. This was chosen due to its value for money as it can perform a lot of functions for a well-priced camera.

1. Place a part on the top of the chute
2. This part will roll down and trigger the px sensor
3. The px sensor will trigger the camera to take a photo of the part in the test area
4. The camera will analyse the part to determine if it's a pass or a fail
5. If a pass the actuator will extend full stroke to push the part out and then retract
6. If fail the operator will need to remove the part and press the reset button.

[29]

which will take a while and throw out the time scale of a planned maintenance activity. Other than that this machine requires very little maintenance due to how basic it is.

7. Discussion and Conclusions

The main differences in solutions throughout this report discuss the outcomes of the project in comparison to what was originally set out. As an overview the project wasn't as successful as hoped for. Although the project was made within budget it lacked reliability proving the chosen method at the start of the project was not fit for purpose and if the project was made to be as reliable as originally wanted in the beginning then the money and time spent on the project would more than likely be greater than would have been spent if the plc method was chosen at the beginning.

Due to the time scale and cost the relays were the most suitable for the project. As long as there weren't any unforeseen problems that occur during the wiring process or the commissioning of the system that will delay this system then this should be really effective for this type of project.

Unfortunately during the commissioning stage of the project the relay logic proved to be very unreliable as when the parts failed it would struggle to reset and take another photo. Another problem was when the parts were passing, the machine was ejecting the part which was triggering the PX sensor causing the actuator to stop moving or retract back in. this would then mean the part wasn't fully ejected causing the next part to position itself awkwardly in the test area which would mean the part would fail. A PLC would have prevented a lot of these problems as timers could have been put in to delay the camera taking an image.

Improvements could have been made on the mechanical design of the chute and test area to prevent jams and parts being mis-positioned in the test area causing them to fail. It might have been better to get a mechanical engineer to design the mechanical layout of the machine with more experience in this field of work.

The camera worked really well and programing it proved very simple. When parts where put in the same place every time it could easily identify the difference between a passed part and a fail part. The problem with the part testing was the part not consistently being positioned in the same please every time in the test area.

The mechanical workings and the relay logic let the project down and would drastically need improving on to make this machine reliable and simple to commission. I would recommend a PLC being added to this system to make it useable.

Although the final system didn't function 100% correctly it showed that the camera can work well for testing for O-rings and more importantly highlighted the importance of good part handling and the precision of part positioning when performing imagery tests.

The project would not have been maintainable if used daily as the camera would need to be reprogramed due to external light effecting the cameras image throughout the day, also it would get too many breakdown and be very cost ineffective.

If there was more time available then this project could have been made to work effectively but the main outcome from this project was to see if it was possible to make a fully automated machine from this concept. Although there are a lot of negatives with this design, the basic concept shows a strong hope for a potential fully automated machine.

The gant chart really helped with the timing of the project as it clearly indicated that the project was going to end up incomplete if anymore time was spent on commissioning which was why the commissioning was ended short.

Reference List and Bibliography

Bibliography

- Agarwal, T. (2013). *Microcontrollers – Types & Applications*. Retrieved February 20, 2016, from <<https://www.elprocus.com/microcontrollers-types-and-applications/>>
- Al-Aubidy, K. M. (2016). *Microcontroller Architecture*. Retrieved February 20, 2016, from <<http://www.philadelphia.edu.jo/academics/kaubaidy/uploads/ES-Slids-lec3.pdf>>
- allaboutcircuits. (2018, 05 01). <https://www.allaboutcircuits.com/worksheets/electromechanical-relay-logic/>. Retrieved from <https://www.allaboutcircuits.com/worksheets/electromechanical-relay-logic/>: <https://www.allaboutcircuits.com/worksheets/electromechanical-relay-logic/>
- distrelec. (2017, 11 30). www.distrelec.ch/en/. Retrieved 11 30, 2017, from www.distrelec.ch/en/: <https://www.distrelec.ch/en/siemens-6es72111he310xb0/p/12571216>
- Electronicshub. (2016). *PIC Microcontroller and Its Architecture*. Retrieved Febraury 21, 2016, from <<http://www.electronicshub.org/pic-microcontroller-architecture/>>
- horner. (2017, 12 1). <https://www.horner-apg.com/en/applications/features/iec-support.aspx>. Retrieved from www.horner-apg.com: <https://www.horner-apg.com/en/applications/features/iec-support.aspx>
- machinedesign. (2018, 05 01). <http://www.machinedesign.com/engineering-essentials/engineering-essentials-what-programmable-logic-controller>. Retrieved from <http://www.machinedesign.com/engineering-essentials/engineering-essentials-what-programmable-logic-controller>: <http://www.machinedesign.com/engineering-essentials/engineering-essentials-what-programmable-logic-controller>
- metroid. (2018, 01 15). www.metroid.net.au. Retrieved from www.metroid.net.au: <https://www.metroid.net.au/engineering/control-methods-plc-vs-relay-logic/>
- MicrocontrollerBoard.com. (2016). *PIC microcontroller tutorial*. Retrieved February 21, 2016, from <http://www.microcontrollerboard.com/pic_microcontroller.html>
- MikroElektronika. (2016). *I/O PORTS*. Retrieved February 21, 2016, from <<http://learn.mikroe.com/ebooks/picmicrocontrollersprogramminginassembly/chapter/io-ports/>>
- Morlote, H. (2013). *Estudio de la estructura interna del PIC 16F877*. Retrieved February 22, 2016, from <<https://cifpn1hectorm.wordpress.com/2013/04/10/estudio-de-la-estructura-interna-del-pic-16f877/>>
- parmley-graham. (2017, november 27). www.parmley-graham.co.uk. Retrieved from www.parmley-graham.co.uk: <https://www.parmley-graham.co.uk/automation/siemens/plc-s>
- plcacademy. (2017, 12 1). www.plcacademy.com/structured-text-tutorial/. Retrieved from www.plcacademy.com: <http://www.plcacademy.com/structured-text-tutorial/>
- plcopen. (2017, 12 1). www.plcopen.org. Retrieved from www.plcopen.org: http://www.plcopen.org/pages/tc1_standards/iec_61131_3/

rohtekautomation. (2018, 05 01). <http://www.rohtekautomation.com/support-software.html>.

Retrieved from <http://www.rohtekautomation.com/support-software.html>:

<http://www.rohtekautomation.com/support-software.html>

siemens. (2017, 11 30). www.siemens.com. Retrieved from www.siemens.com: www.siemens.com

tech-zephyr. (2018, 05 01). <http://www.tech-zephyr.com/programmable-logic-controllers-and-its-advantages/>. Retrieved from <http://www.tech-zephyr.com/programmable-logic-controllers-and-its-advantages/>: <http://www.tech-zephyr.com/programmable-logic-controllers-and-its-advantages/>

P. Horowitz, W. Hill (2015). The Art of Electronics: 3rd Edition, ISBN: 13: 978-0521809269

Al-Aubidy, K. M., 2016. *Microcontroller Architecture*. [Online]

Available at: <http://www.philadelphia.edu.jo/academics/kaubaidy/uploads/ES-Slids-lec3.pdf> >

[Accessed 20 February 2016].

Electronicshub.org, 2016. *Microcontroller Types and Applications*. [Online]

Available at: <http://www.electronicshub.org/microcontrollers/>>

[Accessed 17 February 2016].

Electronicshub, 2016. *PIC Microcontroller and Its Architecture*. [Online]

Available at: <http://www.electronicshub.org/pic-microcontroller-architecture/>>

[Accessed 21 February 2016].

Elprocus, 2013. *8051 Microcontroller Tutorial and Architecture with Applications*. [Online]

Available at: <https://www.elprocus.com/8051-microcontroller-architecture-and-applications/> >

[Accessed 17 February 2016].

EngineersGarage, 2012. *Getting Started with Microcontrollers*. [Online]

Available at: <http://www.engineersgarage.com/tutorials/getting-started-with-microcontrollers> >

[Accessed 20 February 2016].

Microchip Technology Inc., 2001. PIC16F87X. In: *PIC16F87X Data Sheet*. s.l.:Microchip Technology Inc., p. 215.

[.ports/](#) >

[Accessed 21 February 2016].

MikroElektronika, 2016. *WHAT ARE MICROCONTROLLERS AND WHAT ARE THEY USED FOR?*.

[Online]

Available at: <http://www.mikroe.com/chapters/view/64/chapter-1-introduction-to-microcontrollers/>>

[Accessed 17 February 2016].

Morlote, H., 2013. *Estudio de la estructura interna del PIC 16F877*. [Online]
Available at: <<https://cifpn1hectorm.wordpress.com/2013/04/10/estudio-de-la-estructura-interna-del-pic-16f877/>>
[Accessed 22 February 2016].

*God
number of
sources
used*

assignment 2 5007

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

Instructor

39/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22

PAGE 23

PAGE 24

Text Comment. Good Gant chart

PAGE 25

PAGE 26

PAGE 27

Text Comment. More information on the test is needed. How was the camera implemented with the PLC. The PLC program should be evaluated.

PAGE 28

PAGE 29

PAGE 30

PAGE 31

PAGE 32

PAGE 33

PAGE 34

Text Comment. God number of sources used

GRADING FORM: CSG-RUBIC

SEAN JONES

0

RESEARCH-INFORMED LITERATURE



Research is evident, but it does little to tell about the background for the project. Also not all the refs

10/20

have been used in text. Research and introduction - 5/10
References - 5/10 - not all have been used in text.

KNOWLEDGE AND UNDERSTANDING

 Specification is too brief. There needs to be a more technical breakdown of what is required and advancements that are possible. -4/10
Solutions have been considered. There is evidence of alternative methods with test results. The chosen solution is the PLC with the camera, but there is not much on the PLC implementation and configuration. - 4/10

8/20

ANALYSIS

 Testing and results are evident, but isolated to the camera. PLC implementation is missing. There could be a short demo program at least with test results. There seems to be more test carried out on the alternative microprocessor method. - 4/10
Conclusion is weak - 4/10

8/20

PRACTICAL APPLICATION AND DEP.

 3 solutions considered with one alternative tested. - 2.5/5
Final solution implementation is partial because the PLC part has not been covered properly. - 4/10
Quality of work is reasonable. However there are parts missing i.e. PLC . - 2.5/5

9/20

SKILLS FOR PROFESSIONAL PRAC.

 Report lacks structure. Some topic areas are in the

wrong places.- 4/10

4/20