**201AEE Embedded Microprocessor Group Project**

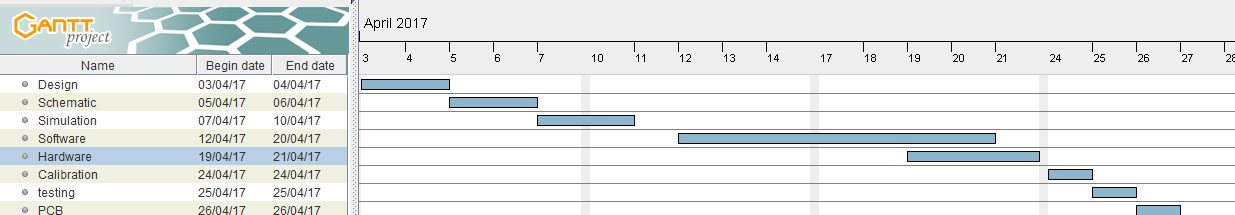
**Names: Alexander Moses 6277932**

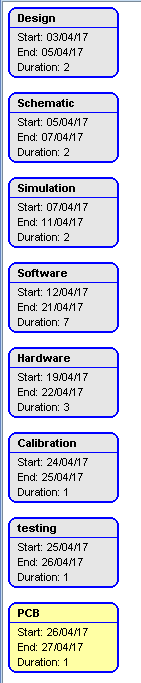
**Sam Stretton 6660721**

**Chantal El Mokdad 6120166**

**Course: Computer Hardware and Software Engineering.**

**Task:** The objectives of this task was to create a robotic vehicle which has the ability to perform several tasks, the First requirement is that the vehicle must be able to perform a 360◦ central rotation in a clockwise direction, another requirement is that the vehicle must be able to pick up if there are any objects in its way, (within the range of 15cm to 35cm, anything out of this range should be overlooked). After the sensor has picked up and allocated the objects in the specified range information about the distance between the sensor and the object itself. Using Bluetooth a report on the findings (scanned objects and the distance between each object and the sensor) should be sent to an output terminal of choice (PC, phone, tablet etc.) and finally the vehicle must be able to perform movements coded for by the user, normally it would be for the vehicle to move towards the detected object.

**Project management**

**<- Pert Chart**

Project management includes all the process/ phases of a successful project this includes everything starting from initiation all the way to the closing of the project. For a project to be successful, several aspects must be considered, the most important are time, cost, scope and the overall quality of the outcome. For our project to reach its final stage it must go through six phases first being setting the goals, the goals of this project is to build a vehicle with the ability to rotate in a clockwise direction and to detect any object within the range of 15-35cm it must also be able to send a report to an output terminal of the users choice, the second phase is the initiation stage this is the starting up of the new project (it could be seen as the foundation) it is the main idea. The initiation stage is followed by the project planning stage this includes the plans of how the work will be carried out, a Gantt chart is normally used to set a time scale, this also includes the resources and cost criteria etc. project execution is the next stage this is the phase where the actual product is created (the product must successfully deliver the wanted outcome. The fifth stage is the project monitoring and control this is where we test the vehicle is actually doing what it is supposed to do (comparing it to the goals set in the first phase) and then lastly, Project closure and this is the end of the project were we are satisfied with the outcome and we can actual deliver the actual project.

**Health and safety**

There are several regulation and laws that have been put in place to ensure that workers are in the safest conditions that the could be in to ensure minimal harm if not none. These regulations must be enforced by both the facility owners (providing specialised equipment, training etc.) as well the person themselves (by actually following the rules and taking certain precautions). The workplace Regulation was put in place in 1992, this act covers the foundation of health and safety which applies to most workplaces (this act excludes workplaces such as mines construction sites, underground mines and ships etc.) since 1992 minor changes were done to this act to ensure practicality and that it is fair and covers all required grounds9 for example for construction sites a construction regulation was adapted in 2007 etc.)

What these acts ensure is that employers ensure that the workplace is a safe and practical as possible for employees. And as for commercial businesses/ properties they have to ensure the safety of not just their employees but for individuals who use their facilities, so during this project the lab should be a safe workplace the facilities must accommodate all (including people with disability), the lab must also be appropriately ventilated so that fresh air going into the lab, so it must have an appropriate number of windows/openings so that the equilibrium of fresh air is maintained. In the Lab, the windows must only open a certain amount as it is situated on the second floor and could be seen as extremely dangerous so they should lock after opening a certain amount. Also, this means that the equipment are tested to ensure that they are safe to use, they are constantly maintained and checked to reduce the chance of fault occurring, and that they can’t cause harm to the user. The temperature should be at least around 13 degrees Celsius, however, people’s preference can actually end up altering this figure. During this project, we should work reasonable hours in the lab this means that we should have regular breaks to ensure minimum health damage such as eye strain/ tiredness, this makes means that we are not overworked and that we are comfortable as we have time to rehydrate etc. also before the project we carried out certain tasks which can be seen as training as we have been both lectured as we as been taught practically about all the equipment that we will use so we are fully aware of the correct way to use them to ensure that we don’t create a hazardous/ dangerous environment for other students. Also during all available lab session there is always a certain number of staff (to ensure the number of staff to student ratio is reasonable) the staff are constantly supervising to ensure that everyone is working in an appropriate manner and no danger to other student occurs, they are also always there to answer questions and advice. Another duty of the university is the type of seating etc. for example as we sit on the computer for a number of hours a day the chairs used should comfortable to stop back and neck strains from occurring. Also, staff member usually monitors the number of student in the lab at a time to ensure that overcrowding does not occur and the students have enough space to move around freely. The lab itself must be clean so the hygiene must be up kept and this includes everything from the ceiling, walls, furniture, and floors (things on the floors are trip hazardous and are taken very serious). Lastly, rules such as no drinking/ eating in the lab are put in place seen as a lot of electrical equipment are found there and contact with water can be extremely hazardous.

**Project risk assessment**

|  |  |  |  |
| --- | --- | --- | --- |
| Hazard | Probability of occurrence | Cause | Prevention |
| Neck/back pain | 6/10 | Sitting for long periods of times on uncomfortable seats | Appropriate seating should be provided if feet could not be placed flat on the floor a foot rest should be provided |
| Eye Strain | 8/10 | Staring at a computer screen for long periods of time without having breaks to rest eyes. Or even looking at smaller things (having to squint). | Take regular breaks allow eyes to rest without looking at the screen or add any pressure. |
| Headaches | 7/10 | Long periods of time focusing without breaks causes headaches also lack of food/water could contribute to a headache. | Take regular breaks to ensure you are not overworked and ensure that you constantly hydrate. |
| Tripping | 5/10 | Wires on the floor or even overcrowding in the lab can be seen as a hazard. | Ensure floors are clean and everything is stored appropriately and wires are tucked away and not in pathways. |
| Overheating | 3/10 | Overcrowding or overuse of electronic can produce a lot of heat and if the temperatures are not regulated and monitored this can even cause some to faint. | Ensure that the lab has a reasonable number of people in it, not too much as the body heat can be excessive also ensure the heating is at an appropriate temperature depending on the weather also ensure appropriate ventilation is available (e.g. Windows) to allow fresh air is constantly entering the room. |
| Tiredness | 8/10 | Lack of ventilation reduces the amount of oxygen in the room cause tiredness, also working for long periods of time cause tiredness and increases the risk of accidents. | Ensure appropriate ventilation is available and also have regular breaks where food and water are consumed to ensure that the body is not dehydrated. Also, rest breaks should be regularly taken. |
| Electric shocks | 4/10 | A short circuit could occur or even an exposed wire that is connected to a power source might come in contact with the person also a fault can occur in the equipment used. | Ensure the power is not kept on when not required. Also, ensure wires used are in good condition and that all equipment are regularly test |
| fire | 4/10 | Sometimes some of the equipment can cause a fault and short-circuit and cause a fire to start. | Ensure all equipment has been tested and use appropriately also have fire extinguishers available to ensure if this hazard occurs it can be put out quickly. |

**Vehicle components**

Unipolar stepper motors differ from a normal stepper motor as it operates with one winding and they have a center tap with each phase. The winding itself consists of a simple commutation circuit (arrangement of the magnetic pole can be reversed without actually having to change the direction of the current itself). In the magnetic field, each of the different directions switches on a certain section of the winding.

Proximity Sensor- a sharp proximity sensor is a type of analogue sensor which has the ability to detect objects at a specific distance (range) away from it. In this project, we needed out the vehicle to detect any objects between the ranges of 15-35cm away so this type of sensor was the most practical.

ULN2803 Darlington driver itself is a high current as well as a high voltage transistor, it is made up of eight NPN pairs that have an output with a high voltage it also consists of clamp diodes which enable it to switch the between the loads which are connected to the output

, this driver is normally used when high power loads are being driven (normally used in things such as motors, relays etc.)

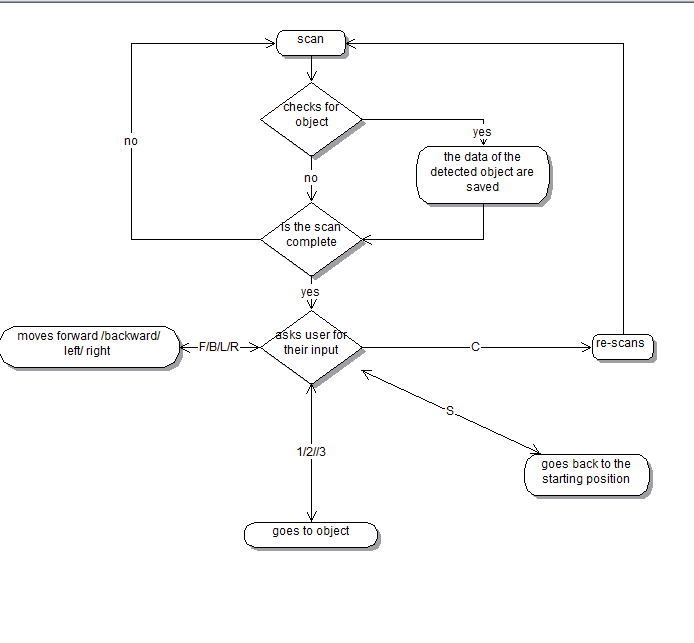
Communication System allows data transmission between different terminals, this is required in this project as we need to transmit data from the sensor (a report on the objects detected including distance etc.) to the output terminal of choice in this case it will be a phone also we will use it to allow us to use the phone to send instruction to the vehicle. What we used to achieve this function is a Bluetooth module.

Resistors are very common aspects of electronic circuits, they are used to implement the electronic resistance in the circuit. Resistors have several functions which include reducing the current flow in the circuit, adjusting signal levels among other uses too. During this project, we used resistor with the value of 10k

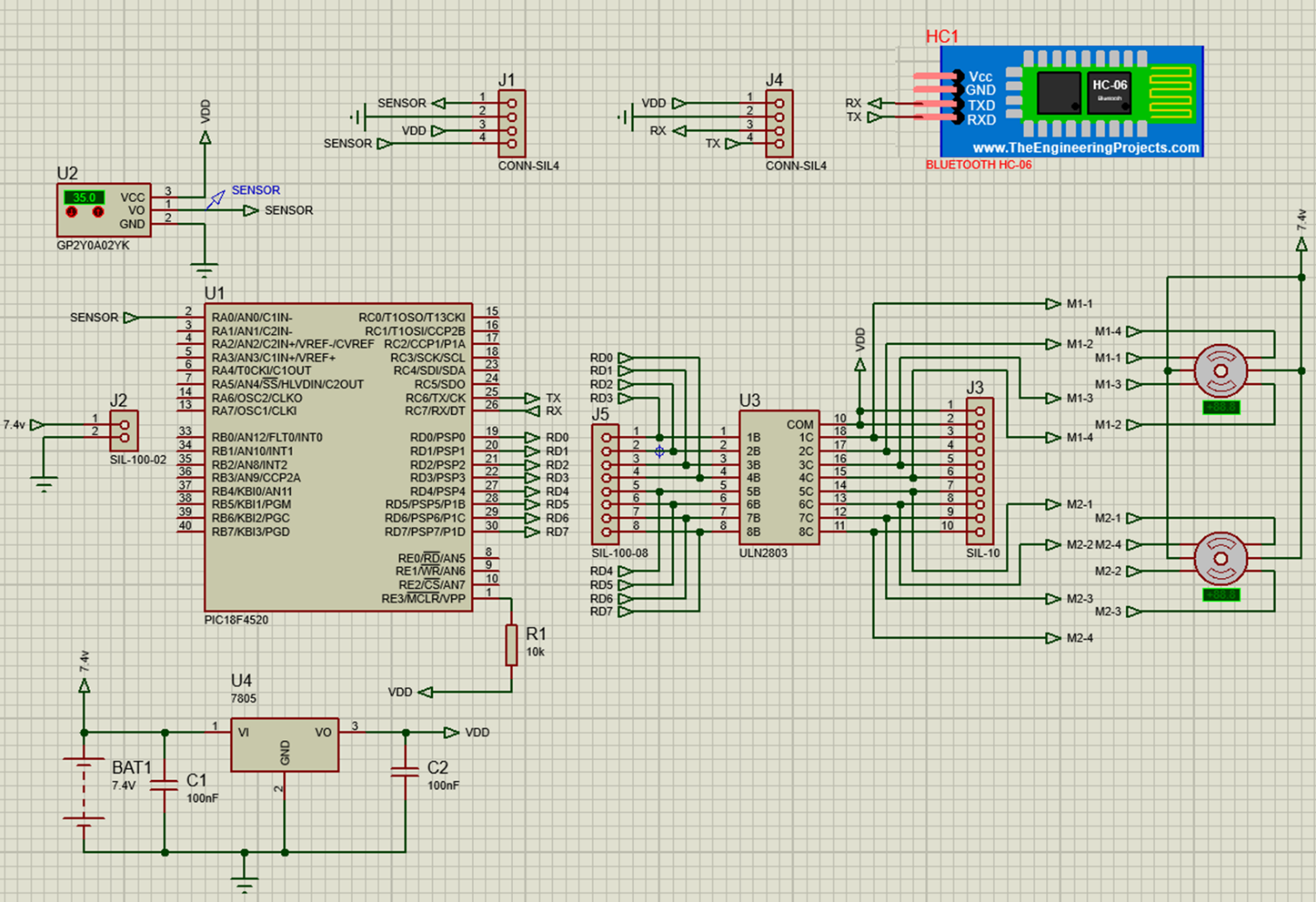
Capacitors are electrical components that have the ability to store electrical energy in a circuit, the main purpose of a capacitor is to be able to enhance this feature in the different particle applications.

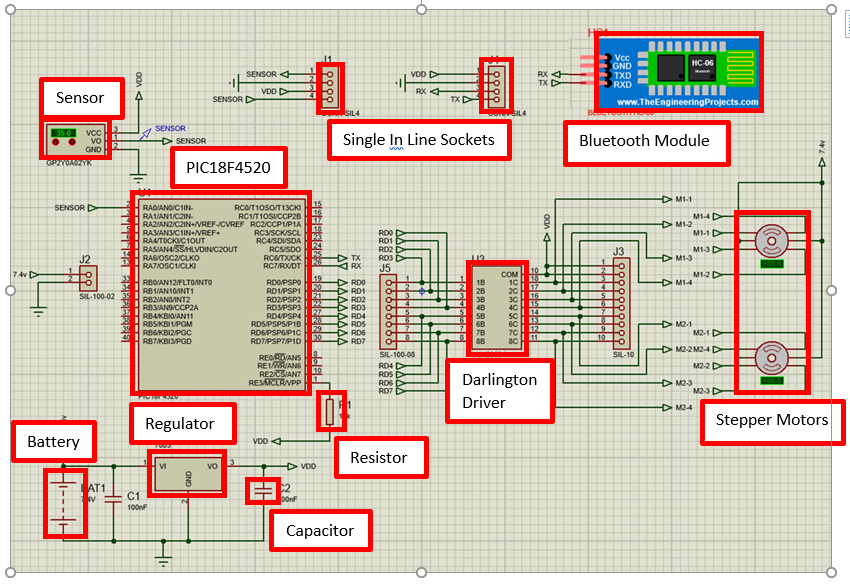
Battery (lithium Polymer Battery) is the power source allowing the whole circuit to function the power source that we used has the value of 7.4 volts

**UML Diagram**



**Schematic**





Proteus Simulation

https://youtu.be/X6qseKYPWMY

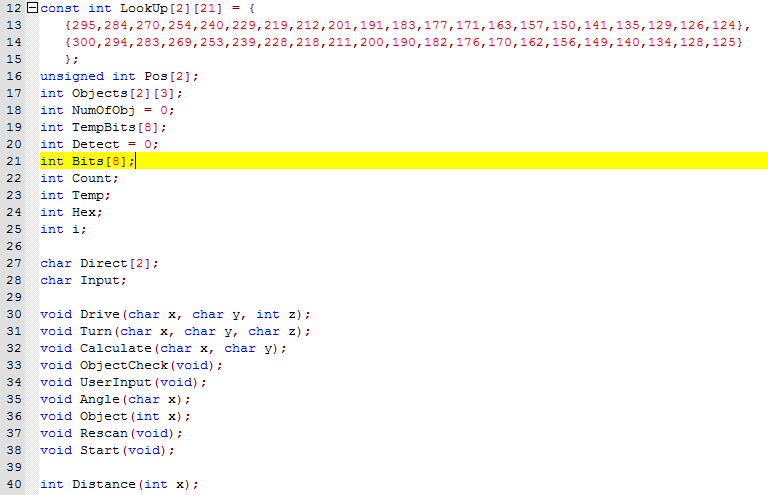
**Code**

The ‘LookUp’ variable holds the upper and lower ranges for the ADC values for each CM, as the voltage values is a logarithmic graph, it’s easier calculating the values for each CM and using a lookup table to figure out the distance

v

The declaration of all the functions used throughout the code

The declaration of all the variables used throughout the code



The ‘main’ function is used to configure the ports, the ADC and the terminal. It also runs the initial scan function (‘rescan’) and contains them main loop for the car to run indefinitely.

The main loop, of which it calls the ‘UserInput’ function which in turn asks the user for input

v

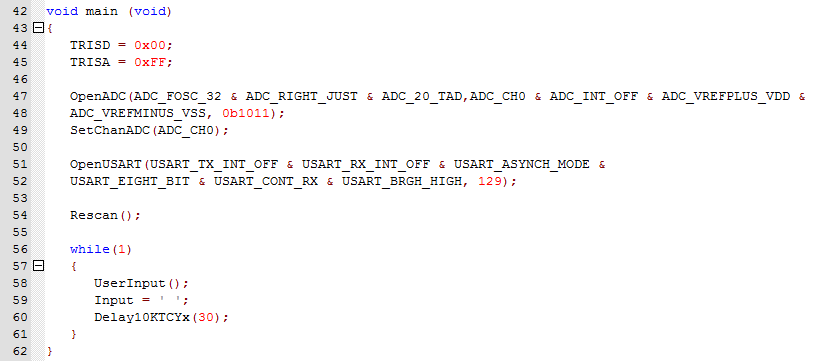
Terminal Configuration

Port configuration and ADC configuration

v

v

v



The ‘Calculation’ function is used to calculate the required hex value for the next step in the cycle of the stepper motor. The stepper motor works in full step operation, each of the 4 steps have a hex value required to turn to that step ( i.e step 1 is 0x0A, step 2 is 0x06). This function takes the previous hex value and turns it into the hex value required for the next step using Boolean expressions. Four Boolean expressions for step 1 to 4, and four expressions for steps 4 to 1 (forwards and backwards). When the function is called, 2 input arguments are used, both arguments are either ‘F’ or ‘B’ for forwards or backwards, the first input argument is for the first wheel, the second is for the second wheel, if ‘F’ it uses the first four Boolean expressions, if ‘B’ it uses the second four Boolean expressions.

The Boolean expressions that convert the binary from LATD into new binary values required for the next step

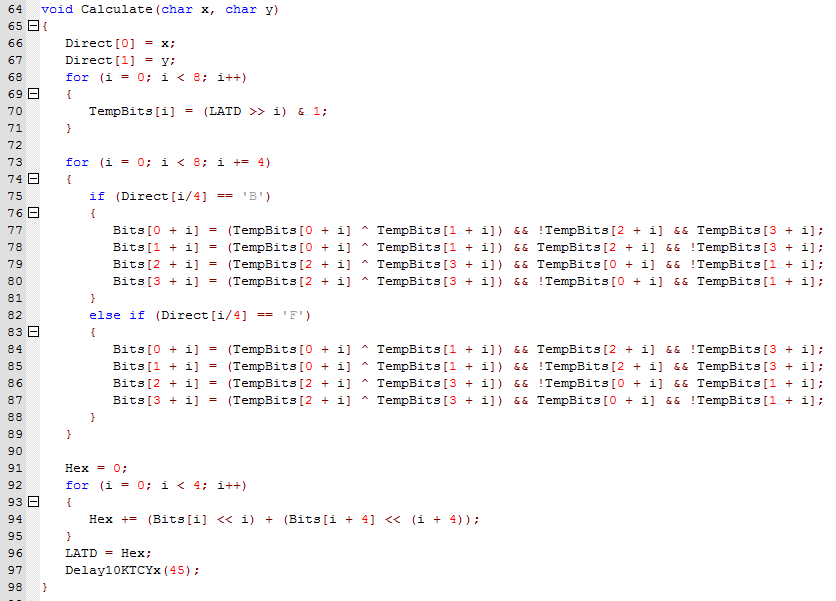
Turns the hex value from LATD into binary and stores it into an array

Turns the newly converted binary into hex and outputs it to PortD

v

v

v



The ‘ObjectCheck’ function is used during the ‘Rescan’ function, is converts the analogue signal from the sensor into a digital signal, checks if that digital signal corresponds with any value within the ‘LookUp’ array. If the distance is indeed between 15cm and 35cm then the object information (distance and angle) is saved to an array. This happens three times (three objects).

Checks if the object is between 15cm and 35cm, if there has been 3 objects or not so far, and whether or not it’s the same object as one previously recorded.

Once recorded, a variable (‘Detect’) becomes 1, and only becomes 0 once it no longer detects an object between 15cm and 35cm, this is used to make sure the same object isn’t scanned twice

v

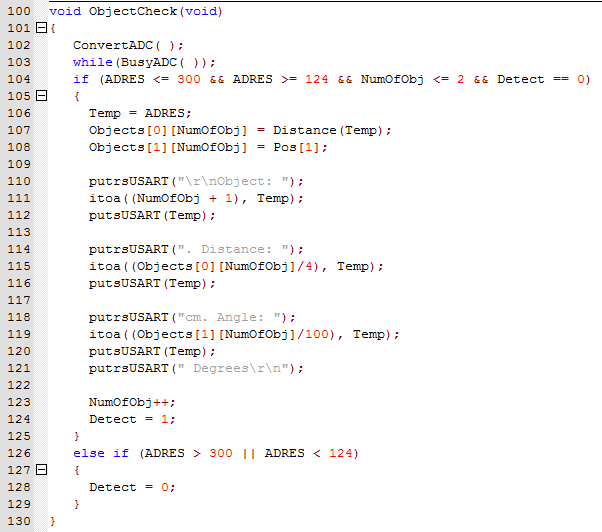
Outputs the information to the user via the Bluetooth terminal

Records the object information to an array

v

v

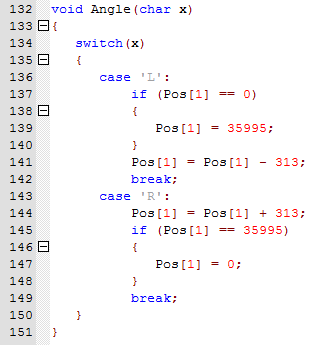
v



The ‘Angle’ function is used to calculate the current angle of the car when it turns.

The switch case is used to determine whether the car is turning left or right. It then calculates the current angle of the car compared to the starting position. The value of 313 is used because the car turns 3.13 degrees every step, and the PIC doesn’t have a floating point register and so I multiplied the value by 100 to minimise the rounding error.

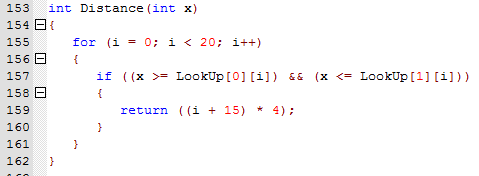
v



The ‘distance’ function is used to calculate the distance of an object by using the ‘LookUp’ variable

It loops through the 20 values (for each cm between 15 and 35). It compares the converted analogue value to those in the lookup table and returns the distance of the object (times 4). The reason its times 4 is because 4 steps of the car are 1 cm, I have done this so it’s easier for the car is drive to an object precisely

v



The ‘UserInput’ function asks the user for an input, and that calls the corresponding function related to the input.

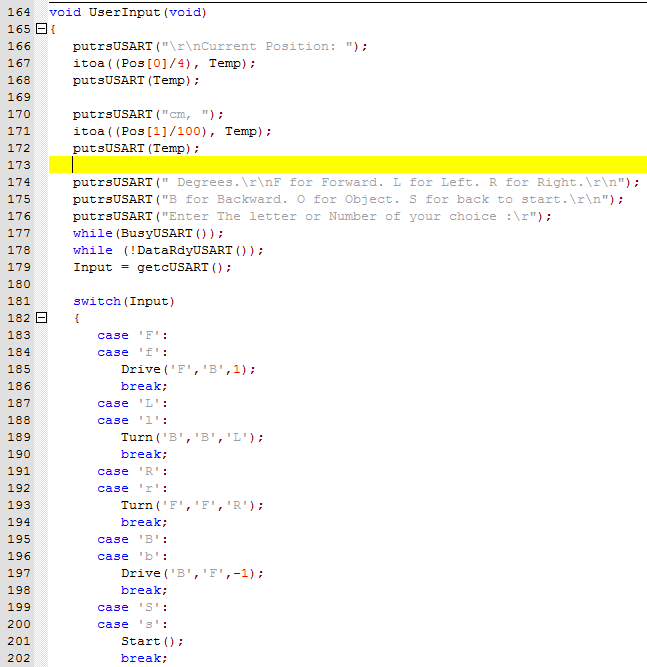
Asks the user for an input. ‘F’ for forward, ‘B’ for backwards, ‘R’ for right, ‘L’ for left, ‘S’ to make the car go back to the starting position.

Outputs the current position of the car in relation to the starting positon.

v

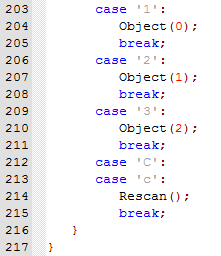
v

v



v

‘1’, ‘2’ or ‘3’ to make the car go to one of the three objects. ‘C’ to make the car rescan the area for objects.



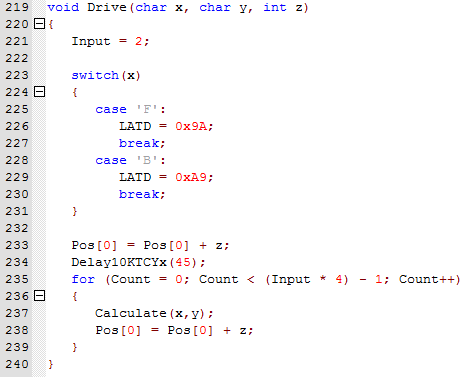
The ‘Drive’ function makes the car move 2cm either forwards or backwards depending on which the user chooses.

Updates the positon of the car, and makes it drive the remaining distance required it to have moved 2cm.

v

As the ‘calculation’ function takes the hex value in LATD and converts it, it requires me to input the first step so that it can be converted. The switch case outputs the first step of either forwards or backwards.

v



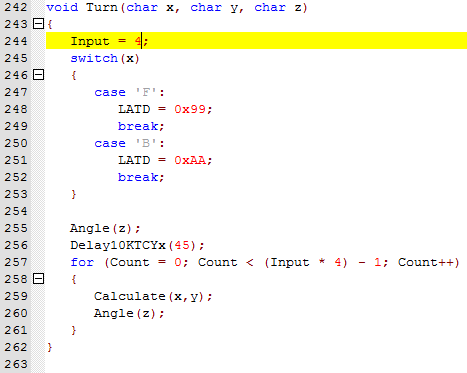
The ‘Turn’ function turns 50 degrees either left or right, depending on the users input.

Updates the angle of the car, and makes it turn the remaining 47 degrees required to make it turn the full 50 degrees.

Like the ‘Drive’ function, the first step of either left or right must be outputted to the port.

v

v



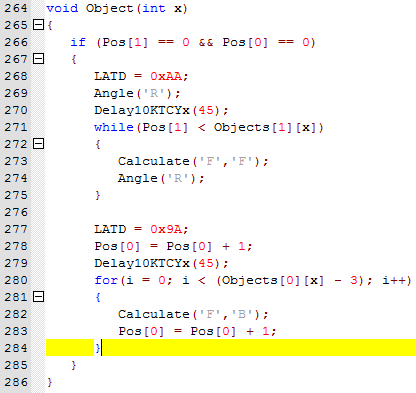
The ‘Object’ function drives to the object that the user inputs (1, 2 or 3).

The car then drives to the distance of the object (minus half a cm to make sure it doesn’t drive into it)

v

The car turns to the angle of which the object is located.

v



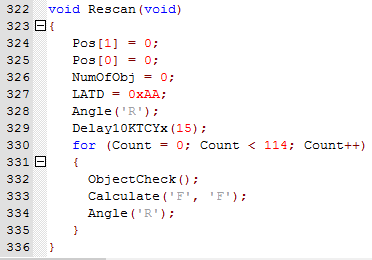
The ‘Rescan’ function makes the car rotate 360 degrees while scanning for objects. Its designed so that this function can be recalled at any time and thus making the car rescan

The car rotates 360 and calls the ‘ObjectCheck’ function to check if there’s an object between 15cm and 35cm

Resets the positions and the number of objects so the car is reset.

v

v



The ‘Object’ function drives to the object that the user inputs (1, 2 or 3).

The car then rotates so its facing the same direction it did at the start.

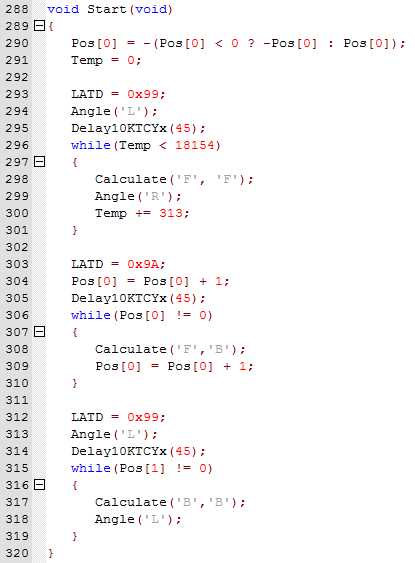
Car drives to the starting position

Makes the car face the starting position

v

v

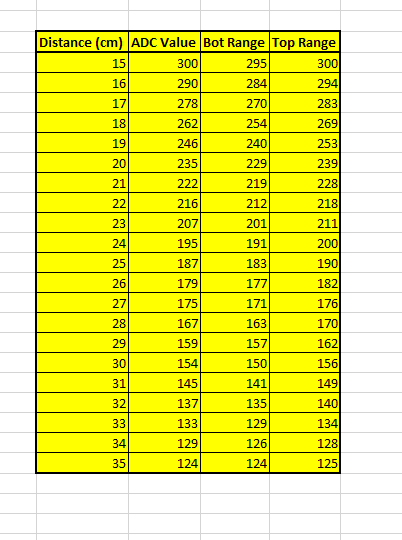
v

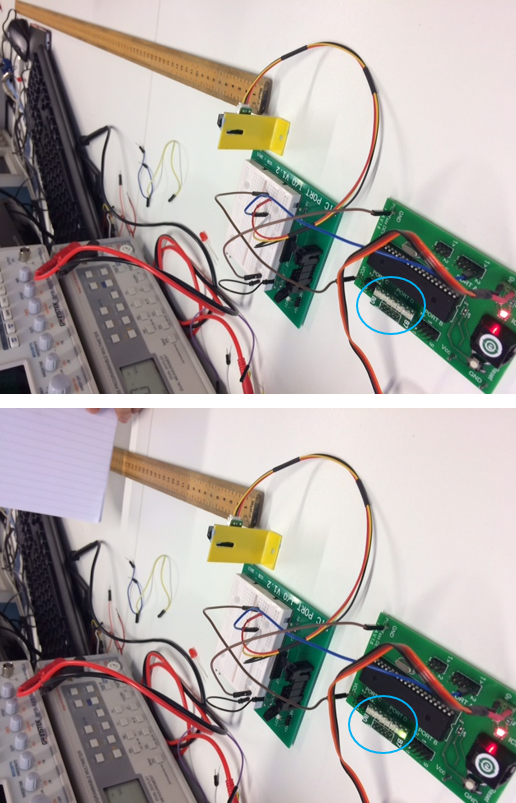


**Sensor calibration and testing**

Sensor calibration is a technique used to enhance the performance and the accuracy of the sensor itself, how this is done is by reducing or removing any of the structural errors in the output, after we were happy with the results we had to test that the vehicle is functioning in the way it is expected to, to do so we place objects within the appropriate range and run the vehicle( it starts to rotate) after a few alteration we were able to get the vehicle to produce a report with accurate data about the positioning of the object.

The data that is provided by the sensor is in analogue form, to make these values readable what we need to do is convert all the analogue values in digital form. The table below shows the range distance that we are using (which is 15cm to 35 cams) the ADC column is the digital value that is collect from each of the specific distances and the third and fourth column are the ranges in which we would consider the object is a specific distance away, to further elaborate if my value fell between the ranges of 295 to 300 I would know that the object detected is approximately 15cm, and if it falls within the ranges of177 and 182 I could assume that the detected object is 26 cm away.





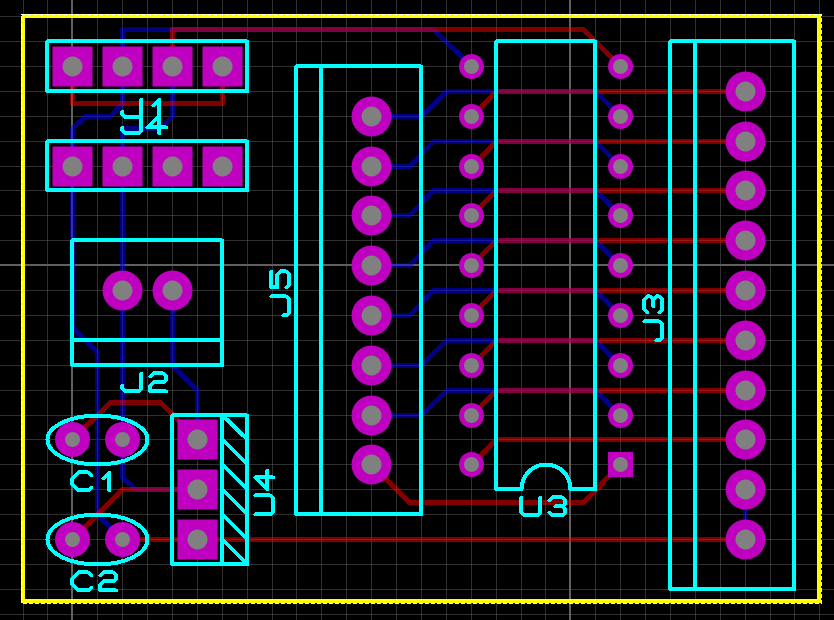
**Project components and Price Analysis**

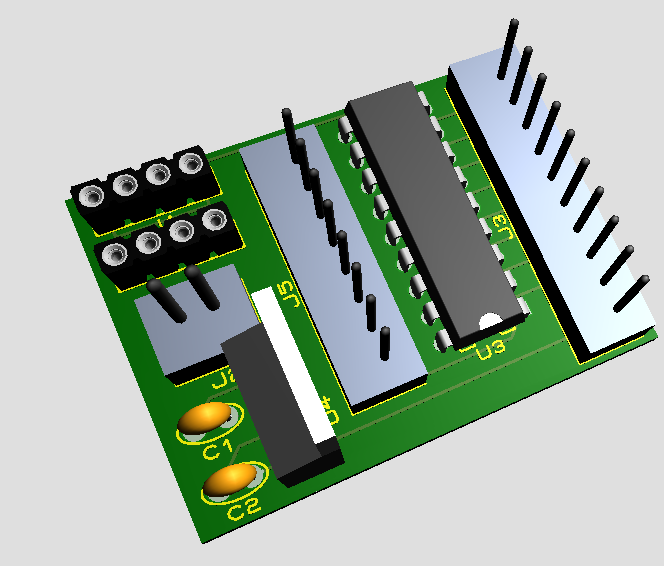


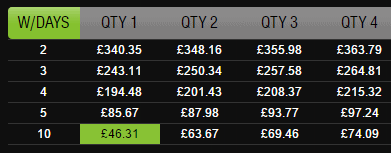
(Proteus stopped working, couldn’t update bill of materials)

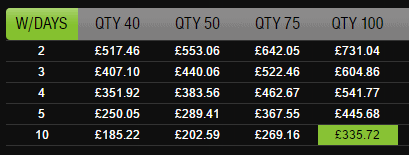
£31.43 + £26.03 (per stepper motor) + £0.32 (extra socket) = £83.81

**Printed circuit board. (PCB)**









**Conclusion.**

Chantal

During this project a variety of skills were tested we had to put all the software knowledge we have gained over the course into practice, first, we had to do a lot of research to enable us to create an appropriate plan of how this task will be done. Also, we had to use a lot of different software to create a Gantt chart, UML charts, schematic PCB designs etc. all these stages were required to make the execution of the vehicle a lot simpler and to minimize the number of errors that can occur. Overall, I think this project went smoothly as our final result (the vehicle) was able to meet all the requirements, first it was able to make a 360 degree rotation and it has the ability to detect any object that was in the 15cm to 35cm range and it successfully sent a report of the location of the object etc. and we were able to control the movement of the vehicle we were able to make it move towards the objects that were detected it was also able to rotate all the way back to the starting position and then move forward, backwards, left and right so overall it did meet all the requirements, and we did follow our original plans, we did stumble over some issues at first after building the vehicle we had to test that it was able to do an exact 360 degrees rotation, however, it was falling slightly short of 360, and the object was not very accurate after looking over it we figured that it was due to a broken sensor after replacing those issue were no longer occurring also the port D was broken and needed to be replaced. And lastly, we attempted to use mathematics to add some features (which included the vehicle driving from one object to the other) however this was not possible seen as the PIC that we use ( PIC18) doesn’t accept floating point ( newer models such as PIC 24 and 32 do)

Seen as we worked in Groups of three we each was assigned as a leader to a section of work with the support of the other members, I was the leader in the first of research, planning, analysis etc., I assisted with the hardware aspect and testing/ calibration.

Alex

My role within the group was mainly the programming of the PIC, which in turn included the design of the logic of which the car operates (UML diagram), and then bringing that design to life by coding it. Upon completion of the prototype code and once the car itself was built I had to calibrate the sensor (and test it) and then make any necessary changes to the code to bridge the gap between simulation and physical hardware. Aside from the programming, I helped the others complete their tasks (building the car and the design).

Once I completed the first version of the prototype code I felt that the way the stepper motors were controlled to be inefficient and unnecessary and so I set out to change it. The original way of operating the stepper motors was to define each of the four steps for each motor (so eight steps in total between the two). This method was inefficient. For each instance of a For loop - to control the motors – each step had to be included, plus with delays and checks it meant it used a lot of code just for one for loop and there were originally roughly 10 For loops. Instead, I used Boolean expressions to transition from one step to another. As a step is just a fixed value (10 -> 6 -> 5 -> 9 -> 10), all that would need to be done is the first step to being outputted and then from there it takes the step, runs it through a series Boolean expressions to get the value required for the next step. Doing so reduced the amount of code by just over 200 lines, and reduced the program memory used by 1000 bytes, however, there was a slight increase in data memory used.

This combined with a few extra ways to reduce the amount of code resulted in the final version only having 336 lines despite it having the extra functionality to what was required. The ‘new’ version was before the car was built and so small segments of code had to be added to it to make it work thus the final version has more lines of code and uses more memory.

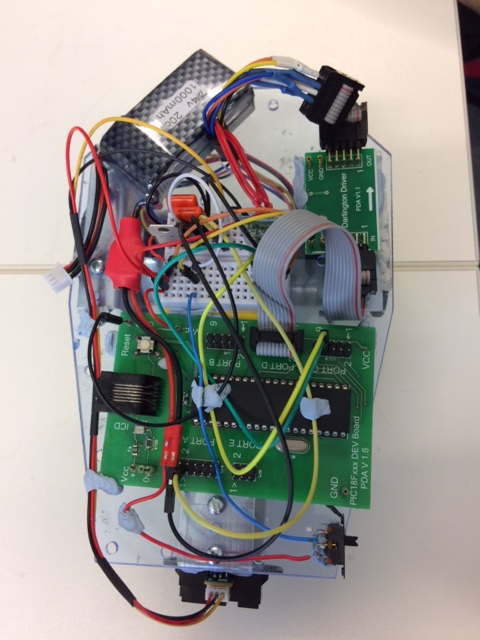
Below is the number of lines of code and memory usage of the old version, the new version, and the final version.

|  |  |  |  |
| --- | --- | --- | --- |
| **Number of lines of code and memory usage** | | | |
| ***Version*** | ***Lines of Code*** | ***Program Memory used (bytes)*** | ***Data Memory used (bytes)*** |
| Old | 528 | 3561 | 427 |
| New | 310 | 2512 | 452 |
| Final | 336 | 3010 | 446 |

While building the car, we ran into a few hardware problems. The first of which was the sensor, which turned out to broken and thus had to be replaced. This was determined by the fact the sensor gave us fluctuating values with two values never being the same. The second problem was that the Port D on the PIC was broken and so the PIC itself had to be replaced but doing so created another problem. The RJ45 port on the PIC was loose and so programming it was a bit difficult but possible and so the replacement of the new Pic wasn’t necessary. Finally, the last problem we had was that the batteries were draining rather quickly, with the car being on all the time and so we added a toggle switch so that we could turn off the car without unplugging the battery.

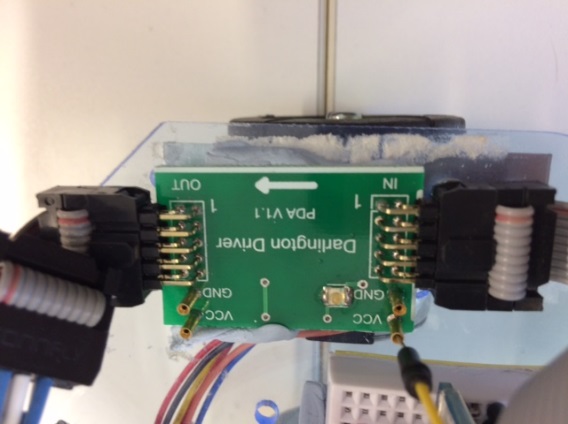
I attempted to use mathematics (cosine rule) to calculate angle and distance of one object to the other to allow the car to drive from one object to another and then back to the start. The PIC18 does not have a floating-point register and so this feature was not possible in this pic. The solution would have been to use a better model (PIC24 or 32) however this feature was not required and so the idea was scrapped.

Apart from all the problems and setbacks, the car operated successfully and meets the requirements. The final functionality of the car includes: a 360-degree rotation at the start of which it scans for objects; outputs a report of the objects to a device connected via Bluetooth; the ability to drive forwards, backwards, left and right; the ability to drive to one of the three objects; the ability to drive to the starting position and the ability to rescan for objects.

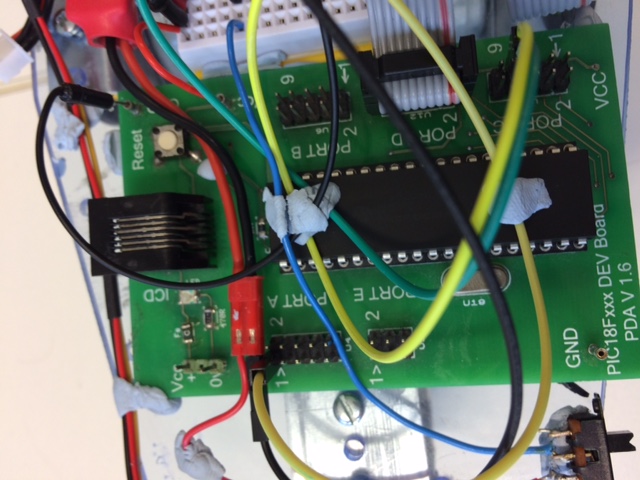
Sam Stretton

My role within the group was the design of the car and the build of the car. This went very well and I added lots to the design to help the ergonomics to ensure that you can pick it up easily and that it is easy to change components. It was also very important that the design could hold up to abuse as it would need to be picked up and moved a lot during testing.

My car design is the picture to the right. I made sure that all the components are cable routing were planned. I will give a few examples of this:



This is my Darlington Driver, there were two drivers I could have used, after trying both I realised that the breadboard version wasted lots of space and caused lots of unnecessary wiring and convoluted the design.

As you can see, on the picture to the right I ensured that the wires didn’t get in front of the RJ11 connector as we were having problems with the connection falling out. I also moved the power wire around the reset switch to make sure it was easily accessible so we could reset the car at any time. I also added a switch to the bottom of the design to make it easier to turn the car on and off. As well as that I blue tacked down the wires that were high up so that the wires didn’t look messy while making sure that components could be swapped easily. I will finish by mentioning that I made sure the battery was at the back and quickly hot swappable.

My main role was hardware but I helped Alex with the software by problem-solving errors and coming up with ideas such as using the cosine rule to go from one object to another, unfortunately. This didn’t work because the PIC we used didn’t have a float register. Other problems I helped work around were the numerous components that stopped working, for example, the port on our PIC didn’t work and we had to work out what was wrong and swap the PIC.

To conclude our project met the specifications which were to. Turn 360, scan 3 objects and complete further actions from the user.

