AUTONOMOUS WIRELESS CONTROLLED ROBOTIC ARM TO PICK UP AND DROP TINY OBJECT

Justin Zhurawel

Bachelor of Engineering (Honours)
Mechatronics Engineering

Department of Mechatronics Engineering
Macquarie University

November 13, 2017

Supervisor: Prof. Subhas Mukhopadhyay

ACKNOWLEDGMENTS

I would like to acknowledge my family, first and foremost for all of their support during my engineering degree/thesis so far, having taught me that hard work and determination pays off along with the many hours they have put in to make me the person I am today, to be able to complete a Mechatronics Engineering degree. A thankyou to my girlfriend Holly who has helped me greatly with time management and my research. Thank you to Prof. Subhas Mukhopadhyay for his help explaining engineering concepts I don't quite understand and for his time to help me better plan my thesis project out and to every other Professor I have had for the duration of my Mechatronics Degree.

STATEMENT OF CANDIDATE

I, Justin Zhurawel, declare that this report, submitted as part of the requirement for the award of Bachelor of Engineering in the Department of Mechatronic Engineering, Macquarie University, is entirely my own work unless otherwise referenced or acknowledged. This document has not been submitted for qualification or assessment an any academic institution.

Student's Name: **Justin Zhurawel**

Student's Signature:

Date: **November 13 , 2017**

ABSTRACT

The precise movement of a Robotic arm is considered challenging, but having to pick up and drop a tiny object wirelessly and autonomously is even harder. The Robotic arm will be required to pick up and drop a tiny object autonomously while also being able to take user input through wireless control. The arm being controlled by wireless communication system poses as a hurdle in terms of data transfer and retention. The main aim of this research is to provide an overview of the methods, specifications and functional requirements of wirelessly controlling a 5 DOF robotic arm. There is certain design challenges that need to be achieved in order for the precise movement of the robotic arm to pick up a tiny object wirelessly. Skills that will be exercised and expanded upon include the use of a 5 DOF robot arm, control system engineering, wireless communication by radio frequency transceivers, MATLAB/Arduino and embedded type coding. Further research and development can be implemented for feedback data from the gripper to further its functionality. The report that follows will outline the experimental methods used and the results obtained through circuit data testing. The main focus of the research is based on the transmission of data and its retention in a control system. As such, further research is needed to design and produce wireless control systems that have a higher degree of accuracy to further improve their functions.

Contents

Acknowledgments iii

Abstract vii

Table of Contents ix

List of Figures xi

1 Introduction xii

1.1 AX-12A Smart Robotic Arm……………………………………………………………….. 1

1.2 Project Overview …………………………………………………………………………… 3

1.3 Baseline Review ……………………………………………………………………………. 3

2 Background Theory and Related Work 5

2.1 Robotic Arm Reequipments……………………………………... 5

2.2 Robotic Arm Specifications ………………………………………………………………….. 6

2.3 Robotic Arm Dimensions …………………………………………………………………….. 7

2.4 Equipment & Facilities ……………………………………………………………………. 8

2.5 Risk Assessment …………………………………………………………………………….. 9

3 Literature Review 10

3.1 Main Components of AX-12A Robotic Arm……………………………………………… 10

3.2 Brief History ……………………………………………………………………………… 13

3.3 Function of Smart Robotic Arm ……………………………………………………………. 14

4 Proposed Work 15

4.1 Methodology & Approach …………………………………………………………………….. 15

5 Experiment Procedures 17

5.1 Main Project Components ……………………………………………………………………. 17

5.2 Project Design……………………………………………………………………………….. 18

5.3 Circuit Creation…….………………………………………………………………………. 22

5.4 Project design testing………………………………………………………………………….. 33

5.5 Full system testing……………………………………………………………………………. 36

6 Results 37

7 Discussion & Conclusion 44

6.1 Discussion ……………………………………………………………………………………. 44

6.2 Conclusion …………………………………………………………………………………… 46

8 Future Work 47

9 Abbreviations 47

Appendix A "Project plan & Gantt Chart" 48

A.1 Overview………………………………………………………………………………………… 48

A.2 Project Plan……………………………………………………………………………………… 48

A.3 Gantt ……………………………………………………………………………………………. 49

A.4 Consultation forms………………………………………………………………………………… 50

Appendix B "Arduino Code" 52

B.1 Transmission Code……………………………………………………………………………… 52

B.2 Receiver Code………………………………………………………………………………… 56

B.3	HC-12 Baud rate change code………………………………………………………………..	57
B.4	FSR LCD test code…………………………………………………………………………	59
References		60

List of Figures

FIG 1 AX12A smart robotic arm…………………………………………..….2

FIG 2 Shows an advanced car building robotic arm………………….4

FIG 3 Specifications of AX-12A Dynamixel Arm…………………..5

FIG 4 Dimensions of AX-12A SMART Robot Arm…………….…..6

FIG 5 Base Plate of Robotic Arm……………………………..…….11

FIG 6 Unimation Robot arm…………………………………….……..15

FIG 7 HC-12 433MHz Transceiver………………………...……….15

FIG 8 Arduino Uno…………………………………………………….15

FIG 9 Arduino Nano………………………………………………..….15

FIG 10 USB to Dynamixel Controller…………………………..……15

FIG 11 Arduino PS2 Joystick…………………………………….…..15

FIG 12 Arduino Pro Mega……………………………………………..15

FIG 13 Transmitting Circuit Configuration………………….……….16

FIG 14 Receiver Circuit Configuration………………………….……...17

FIG 15 74LS24 Line buffer…………………………………………..….22

FIG 16 Arduino Mega…………………………………………………...21

FIG 17 Soldering Pins…………………………………………..….…..22

FIG18 HC-12..23

FIG19 Arduino Nano……………………………………………………23

FIG 20 Bread board……………………………………………………..24

FIG21 Final Controller …………………………………………….….25

FIG22 Receiver Circuit “Original”………………………………..…….25

FIG23 Line buffer……………………………………………………….26

FIG24 Arduino Mega…………………………………………………...26

FIG25 Signal and gnd connector………………………………..……27

FIG26 New final receiving circuit………………………….……...……27

FIG27 FSR………………………………………………………………....28

FIG28 Feedback Cicuit……………………………………………………29

FIG29 Display of gripper force……………………………………….....29

FIG30 Testing FSR……………………………………………………...30

FIG31 Accelerometer test circuit…………………………………….. 30
FIG32 Accelerometer on end effector……………………………31
FIG33 Final 2.0 Setup of receving circuit………………………..…31
FIG34 Loctite bolts……………………………………………………..32
FIG35 Test data 1115200bps 50ms delay………..………………..……33
FIG36 Test data 1115200bps 20ms delay……………………………...33
FIG37 Test data 1115200bps 10ms delay……………………………34
FIG38 Test data 57600bps 50ms delay……………………………...…34
FIG39 Data Stream…………………………………………………...36
FIG40 Test 1……………………………………………………………..37
FIG41 Test 2…………………………………………………………..…38
FIG42 Test 3…………………………………………...…………………39
FIG43 Test 4……………………………………………………………...39
FIG44 Test 5…………………………………………………………………40
FIG45 Test 6…………………………………………………………………40
FIG46 Test 7…………………………………………………………………41
FIG47 Test 8…………………………………………………………………42
FIG48 Test 9…………………………………………………………………42
FIG49 Test 10……………………………………………………………..…43

List of Tables

Table 1 Time Budget Data……………………………………………………………..3
Table 2 Remaining Budget………………………..…………………………………..3
Table 3 Cost Break Down ……………………………………………………………4
Table 4 ENGG411 Research Thesis Project plan……………………………48
Table 5 Gantt Chart…………………………………………………………………….49

Chapter 1

Introduction

The use/need for robotic arms in everyday life and industry is forever growing as their usefulness in a wide range of fields is ever increasing, a AX12A smart robotic arm is a more simplified down version of an industrial robot they use for manufacturing but allows the user to learn the basis of a multi axis robot arm and how they work. The problem that robot arms address is the accuracy, reliability and consistency of their work compared to that of a humans, they have a vast array of advantages in business over humans by the way they can work 24hrs straight and produce near identical quality work on each task they perform, they can also be used to do more dangerous tasks that may be too risky for a human worker to do and most of the time a faster pace which is ideal for a company's bottom line. These are important points within a business as the bottom line of any business is to make money and if a remote arm can do the job better and faster than the equal human counterpart then they become obsolete and the business can either move their job to a new section or can cut back on wages.[1][2]

1.1 AX12A Smart Robotic Arm

The Wireless Control of a robotic arm has led to great advances in manufacturing and industry The aim of this thesis project is to use a smart robotic arm consisting of 7 "AX12A" Dynamixel smart servos including wireless control in order for the arm to pick up and drop a tiny object for a given duration autonomously and with a high degree of accuracy. While implementing the use of feedback data to further increase the system functionality.

Fig 1. AX12A smart robotic arm

The short term goals of this project is to get the robotic arm to perform the given task wirelessly and reliably in terms of picking up and dropping the given tiny object. The longer-term goals are to have the robotic arm performing the same task as in did the last and will do the next, accuracy is paramount in the use of robot arms as when they are used in industry their product needs to be accurate and reliable for the consumer and manufacture. Another goal for the future of the robotic arm is the ever increasing technology used in them, so they can evolve to do a larger number of tasks, faster and more accurately while remaining affordable for business.[5][6] The non-goals of this project would be to be able to perform the given task in a safe manner without endangering the public or user of the robotic arm. Unreliable behaviour and inaccurate work would also be a non-goal for this project moving forward. The main requirements associated with the stake holders in the development and use of a robotic arm, would consist of the customers having training in the safe use of the given product along with the supervisors overseeing the robotic arms be training in a matter to be Able to keep the work place safe and free from and dangers associated with a robotic arm.

1.2 Project Overview

This section contains a short description of the thesis project. The Project Plan is listed in Appendix A.2 along with progress timeline in A.3. This Project is being supervised by Prof. Subhas Mukhopadhyay from the Department of Mechatronic Engineering at Macquarie University. Regular meetings are organised each week to update my supervisor of progress and any problems that have arisen. The layout of the thesis is explored below

1.3 Baseline Review

This project is planned from the 31st July to the 6th Nov 2017. This baseline review is a plan to make sure that all allocated time is used efficiently and effectually. This is broken down into 2 categories “Financial Budget” and “Time Budget”.

Time Budget-

The Table 1.1 below shows the total number of days allocated for this project along with the amount of days consumed.

Allocated Work days	99 Days
Work days consumed	99 Days
Percentage to completion	completed

Table 1 Time Budget Data

Allocated Budget	$300
Amount Spent	$190.10
Percentage spent	63.36 %

Table 2 Remaining Budget

Financial Budget-

The amount allowed for thesis project is $300. A proportion of this budget has been spent thus far, which can be seen in the table below

Items bought for Thesis Project	Quantity	Supplier	Cost
DRI0027 Smart Servo Sheild	1	Core Electronics	$52.40
9V Battery Clip	2	JayCar	$3.95
Polycarb Box	1	JayCar	$14.95
Momentary Push Button	2	JayCar	$2.90
Slide switch	1	JayCar	$1.65
16x2 HD44780 LCD DISPLAY	1	Ebay	$7.77
1N4007 Diodes	20	Ebay	$3.00
140pc U shape Bread Board jump	1	Ebay	$6.10
Joystick Ps2	2	Ebay	$7.06
Nano Atmega328p 16M Arduino	1	Ebay	$7.60
433MHZ HC-12 Transceiver	4	Ebay	$38.00
Solderable Bread Board 400 hole	1	JayCar	$11.95
Nano V3.0 CH340G Arduino	1	Ebay	$7.78
Arduino Mega 2560	1	Ebay	$25.00
Totals	39	Ebay	$190.1

Table 3 Cost Break Down

Chapter 2

Background Theory And Related Work

2.1 Robotic Arm Requirement

The research question in regards to this report is the "Autonomous wireless controlled robotic arm to pick up and drop a tiny object" this will be made possible by using the following skills that can relate the use of robotic arm and control engineering practices along with wireless communication and the embedded codding in MATLAB/Arduino. The task at hand will require a 5 DOF(Degrees of freedom) robot arm using 7 "AX12A Dynamixel smart servos" to pick up and drop a tiny object wirelessly. This will be for a certain duration of time while being controlled through the wireless communications system. Robot arms have great importance in modern day to day life and in industry it is present when the need to perform a heavy and repetitive task or job is needed to be accomplished, this is why industrial robotic arms are exceling at an increasing rate. They are also extremely useful to handle a wide variety of task which may be difficult, dangerous or too repetitive for humans to do consistently. It is important that robotic arms can do such tasks in order to keep human workers safe in the workplace while also increasing the efficiently and production while minimizing the bottom line cost model for a given company[3].

Fig 2. Shows an advanced car building robotic arm.

The use and need for advanced robot arms in manufacturing industry is building at an ever increasing rate with greater efficiently and task capability's. The focused use of robotic arms in the car manufacturing industry is ever increasing, mainly being used for welding the metal frames of the cars together with a high degree of accuracy and speed. Robotic arms are also heavily used in CNC(Computer numerical control) machining robots. Although a CNC machine does not resemble a typical robotic arm as they are usually restricted by 3 axis, where as a robotic arm usually has more axis they can perform similar tasks with relative ease[4]. A 6 DOF robot is the ideal number of joints for a robotic arm as it is able to reach all positions from all orientations in the robotic arms 3D work space.

2.2 Robotic Arm Specifications

- AX-12A Torque / Speed – AX-12A - 15kgf/cm and 59 rpm
- Overall length (from base) - (56.51cm)
- Weight – 1kg (Standard Smart Arm – no upgrades)
- Robot Base height - (2.54cm)
- Base rotation servo is centred in a (10.16cm x 10.16cm)
- Repeatability 2.5mm (Accuracy)
- Each axis has a 300° range with 1024 steps (0.29°/step)
- Initial base angle can be set in 45° increments.
- Precision carbon steel ball bearing turntable
- Integrated pem nuts for quick and easy construction
- Scratch resistant type II anodized coating
- Dynamixel AX-12A (5 axis + 1 gripper)

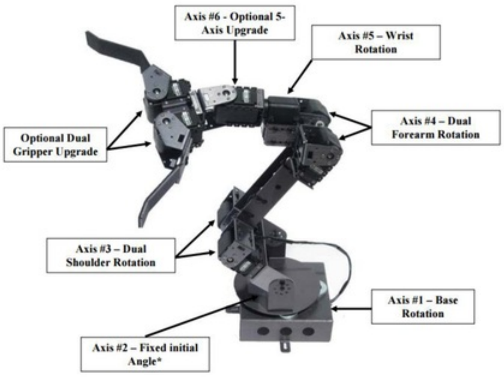


Fig 3 Specifications AX-12A Dynamixel Arm

2.3 Robotic Arm Dimensions

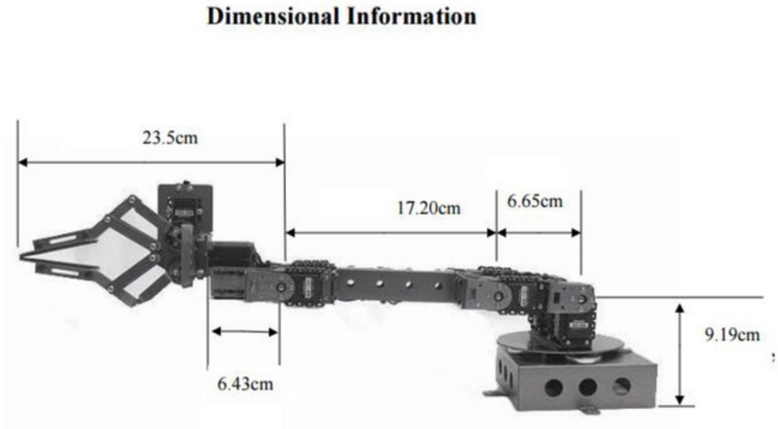


Fig. 4 Dimensions of AX-12A SMART Robot Arm

Dimension Specifications-

- Overall Length – 56.54cm
- Overall Weight – Standard Arm with fixed initial angle bracket - 1kg
- Sensor engineered gripper design
- Built in, adjustable camera / sensor mount
- Slotted gripper paddles for easy sensor attachment
- High resolution, 60 tooth, heavy duty gear train
- 2x (6cm) Standard Gripper opening
- Cross cut, high adhesion rubber for maximum gripper adhesion

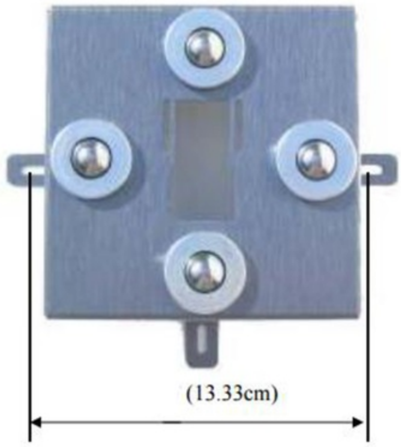


Fig 5 Base Plate of Robotic Arm

Base Specifications-

- 360’ Turntable
- Stainless steel, sealed ball bearing turntable
- 3- integrated mounting tabs for easy mounting to any base
- Base Size – (11.43cm X 11.43cm)

2.4 Equipment and Facilities

The equipment needed for the thesis project consists of the:

* DYNAMIXEL AX12A smart robotic arm
* LCD(Liquid Crystal Display)
* Simulink, MATLAB and Arduino software
* Wireless device HC-12 Transceiver for use of wireless control of the robotic arm.
* FSR(Force Sensitive resistor) for gripper feedback
* Arduino Uno
* Arduino Mega Nano
* Dynamixel Shield
* SN74LS241N Line buffer

Facilities needed would be fabrication engineering rooms which will have the required tools, computers and software in order to configure the robotic arm in the desired way. The costs of the DYNAMIXEL smart robotic arms start from $500 up to $800 depending on the configuration and model. The Arduino Uno/Arduino Mega and Dyna mixel Smart servo shield device costs around $140 combined. Other costs with the project includes the workshop consumables which work out to $50-$100 this is directly related to the improvements to the arm is needed. Software costs that are associated with the project are availability from the university's licenses therefore they have not be put into the project budget. HC-12 Transceivers for wireless communication price is $20 per pair of transceivers larger more powerful transceivers are available for a greater price. The use of the CNC machines and 3D printers will also be needed for this given thesis project along with appropriate software. Total cost minus the robot cost will come well into the $300 limit for this project. All the facility's needed for the thesis project are available within the university labs this results in no need for specialist paid facilities. The finalized design parts consist of 1x Arduino Nano for transmission ,1x Arduino Mega Nano for the receiver along with 2x HC-12 433MHZ Transceivers for the wireless transmission part of the design these are the main parts that make up the design which are also paired with joysticks and momentary buttons that lead to added functionality of the project. The approximate total costs of the finished project was $191. If the fund limit for the thesis project of $300 was overdrawn, permission from the project supervisor would be needed in order to purchase further items.

2.5 Risk Assessment

The risks associated with this project are as follows, the possibility that the arm can make an unexpected movement they may result in harm to the user if it was to strike the face or near the eye region, for this reason eye protection was worn at all times while the robotic arm is connected to power. The risk of electrical shock was possible due to the nature of creating electronic circuit's this was minimized by safe working procedures and making sure there is no exposed power cables when devices are plugged in and turned on. Appropriate care was taken when using mains power for circuit's as there is a potential 240VAC power which can cause sudden cardiac arrest.

Chapter 3

Literature Review

3.1 Main Components of AX-12A Robotic Arm

The controller-

The controller of a smart robotic arm is its command centre which allows all of its parts to interact with each other in order to perform a certain task. This works as a micro controller that allows the arm to connected to multiple systems simultaneously. The control module has a list of instructions that are used in order program a control code. This dynamixel program has a learning capability function so it can take user input and feed it back. Most modern smart robotic arms use an interface that runs a windows based operating system. The more joints a smart robotic arm has the more degrees of freedom it can use. The more simple smart robotic arms usually have three degrees of freedom whereas larger industrial type smart robotic arms use at least 6 degrees of freedom which allows the smart robot arm to reach every position in its 3D work space from every angle configuration, any more DOF is not beneficial[11].

The end effector-

The end effector is the device that the smart robotic arm uses in order to interact with the manufacturing environment, this differs from what application the robotic arm is used for, the end effector is where the working tools are attached. These tools usually consist of but is not limited to welder, spray painter, 3D printer nozzle, cutting tool, gripper. Each of these different tools require their own specific programs but most robotic arm end effectors are interchangeable so that 1 robot can do a wide variety of tasks.

The drive-

The drive is the means of which the joints move these are usually electric motors but can also consist of hydraulic and pneumatic drive systems. The motors are what control the speed and torque of each joint of the robotic arm. They are connected by bars that link each of the motors to create the arm itself. Some Industrial smart robotic arms use a combination of drive systems as each have their pros and cons for a specific task. The electric motors have not as fast movement compared to that of hydraulic but are usually cheaper to produce and run. Hydraulic drive systems have a high amount of strength and speed but are usually come with a higher production cost due to their parts. Whereas pneumatic power is usually quite cheap, there usually require air conditioners to lubricator their internal systems along with motor silencers for their compressors.

Sensors-

The sensors allow the smart robotic arm to have feedback from its 3D workspace and then be able to act according. The sensors get different types of information depending on the specific sensor about its workspace and then sends the appropriate commands to the control unit in order to process its next task or movement. The sensors make the smart robotic arms have a higher degree of accuracy while also making them safer to any potential or unforeseen problems . The robotic arm sensors also allow the robot to adjust its end effector to prevent it from hitting foreign objects or being able to move to avoid an obstacle or person.

Specifications for "DYNAMIXEL AX-12A SMART ROBOTIC ARM" :

- They use open source programming
- The smart robotic arms has features for feedback and position along with voltage, temperature and current.
- The base angle is adjustable.
- 5 different aftermarket end effector options
- Precise control over a 300 degree range for position
- The speed and torque can be adjusted in 1024 increments
- Can turn off automatically if the temperature or voltage is exceeded this is shown by the LED marker.
- The end effector can accept pressure sensors and IR detectors.
- 1 million baud rate serial communication.
- Dual servo design in the second and third effector which allows for better lifting (0.90kg to 1.36kg)
- Open source coding platform
- Compatible with Arduino and MATLAB and many other programs
- Aluminium construction.
- Corrosion and scratch resistance coating.

3.2 Brief History

The need for smart robotic arms devices like the Dynamixel are important for manufacturing and to work with humans has long been a desire. The ideas for robots were first thought up many years ago. The main turning point for smart robotic arm were when they were introduced into the industrial automation sector for manufacturing cars. After a while developers were creating more complex and more useful robotic arms that have a variety of external sensors which aided in their increased accuracy when using such end effectors as grinding, welding and assembly. Robotic arms have been a very useful tool when it comes to manufacturing as it has led to more productivity and a higher degree of quality, while also reduced risk in the workspace. With thanks to "George Charles Devol" one of the first industrial type robotic arms were designed and created. His invention came as the "The unimate" in 1954[6].

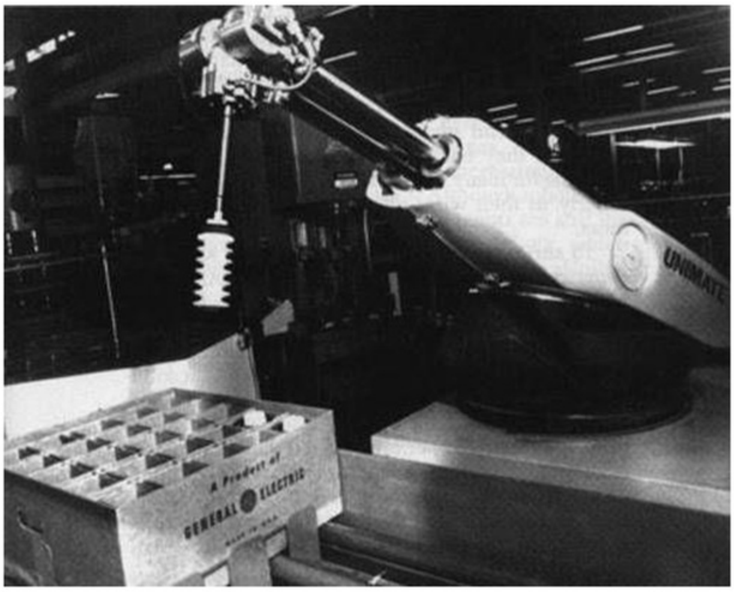


Fig 6: Unimation Robot arm

The first working commercial porotype "Unimate" was made in 1961.
1950s -The introduction of numerically controlled (NC) machines came in.
1970s – Integrated circuits lead to the first industrial style robot arm for manufacturing.
1960-1970s the industrial robotic arms started to gain increasing popularity.
This competition led companies over the world to produce more to meet the growing demand of smart robotic arms for manufacturing.
In 1963 the first 6 DOF "Rancho Arm" was created to aid the handicapped.
In 1968 a smart robotic arm was developed to be able to lift a person and had 12 DOF.
In 1967 first spray painting arm was created.
In 1969 a robotic arm was the first controlled by a computer.
In 1974 robotic arms were able to produce high enough degrees of accuracy which allowed them to be able to make a water pump for the Model T Ford.
In 1975 industrial robots grown by 30% per year up to 1978. This was the turning point that lead to smart robotic arms taking off and commercial manufacturing has never been the same since.
In 1980's Car companies started to use robotic arms as investments because they were cheaper then employing humans to do the same work. General Motors spent more than $41 billion US dollars on new robot technology.In 2014 sales for smart robotic arms worldwide increased by 29%. That was the highest grow in 1 single year since their conception. This history shows the need and many uses for smart robotic arms, they are a valuable asset to any manufacturing business and come with great benefits that aid the company in further advancements in their own sophistication and control technology.

3.3 Functions of smart robotic arm

1. Auto shutdown if unsafe.
2. Feedback like live position.
3. Temperature control
4. Position control
5. Speed control and feedback
6. Voltage control
7. Load feedback and control
8. Cable or wireless connection
9. Emergency shutdown
10. Error feedback
11. High resolution gearboxes for smooth operation.
12. User abilty to configure the margins for speed and torque.

Chapter 4

Proposed Work

4.1 Methodology and Approach

The methodology that was used too achieve the intended goal was by using the AX12A smart robotic arm along with the dynamixel smart servo shield for servo control along with a pair of HC-12 433MHZ transceivers for the wireless component. The AX12A smart arm was connected to an Arduino Uno which had the servo shield connected along with one of the HC-12 433MHZ transceiver, this made up the receiving side of the communication channels. The transmitter circuit consisted of an Arduino Nano which was connected to 2x Ps2 Joystick modules which allowed the Nano to take in the analogue signal from their position, the Nano was also paired with a HC-12 to transmit the position of the joystick to the receiving circuit in order to control the smart robotic arm according. The construction of the robotic was straight forward as it is an off the shelf product. The potential task of 3D printing gripper covers for the end effector was thought out as they do not meet at a nice angle to be able to pick up and drop a tiny object. The steps needed to make sure the project followed smoothly was aided by the design of a budget and time calculator which allowed the project to run on time and budget. Models were also used to better the progression of the project by allowing it to be clearly seen which components would be needed for the designed task. Prototyping was also used to allow the creation of test electronical circuit's on bread boards which aided in their design and refinement due to their plug and play functionality. Testing will be ongoing and be done at different stages of development to make sure that the smart robotic arm is doing its intended task, while also allowing for system refinement . Statistical modelling for system errors in the testing phase will allow the system to be better refined and accurate to given inputs. Any assumptions made in the project undertaken should aid in the systems design and its ability to able to produce the given task outlined by this thesis project. As this type of system already exists in the world the overall improvement and understanding of such a task is the main goal. It is calculated that the given project is achievable in the given time frame and budget. Therefore this project can be justified and therefore delivered.

The approach of this task is to start with research into which items and hardware would be required to produce the desired results, followed by the cost break down of materials needed. Next the abilty for all desired parts to be able to communicate with each other as needed is important for the overall project unity to produce the desired goal. System testing is needed to make sure the project is working as planned. The end goal of the given project is to have the smart robotic arm being controlled by wireless in order be able to pick up and drop a tiny object autonomously. This system can be further optimized for future work to allow the feedback control to the user to aid in more functionality of the system.

Chapter 5

Experimental Procedures

5.1 Main Project Components

FIG 7 HC-12 433MHz Transceiver

Fig 8 Arduino Uno

Fig 9 Arduino Nano

Fig 10 USB to Dynamixel Controller

Fig 11 Arduino PS2 Joystick

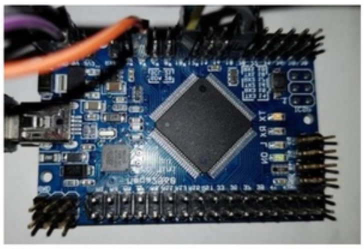


Fig 12 Arduino Mega Pro Mini

5.2 Project Design

The Final project design consists of 2 Arduino devices 1x Arduino Mega Mini model and 1x Nano Arduino. The Arduino Nano is used for the controller as it is a lot smaller than a conventical Arduino Uno but with the same functionally. The Arduino Nano Has HC-12 Transceiver connected to it in order to wirelessly communicate with the receiving circuit. This is done by the HC-12 wirelessly relayed the data being printed out on the Arduino Nanos serial monitor and transmitting it to the receiver in order to control the robotic arm. The Arduino Nano also has 2x Joysticks connected to its analogue in pins in order to gain user input and then relay that to the receiver's serial monitor in order to control the robotic arm. Theses joysticks along with the HC-12 transceivers are power by the 5V out on the Arduino Nanos board. While the board itself is powered by a 9V battery. The Circuit of which can been seen below

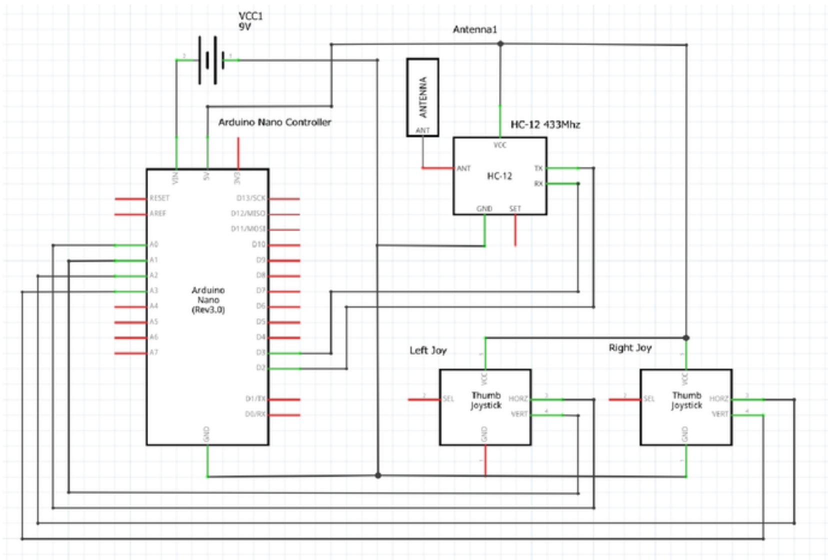


Fig 13 Transmitting Circuit Configuration

The Arduino Uno was originally used for the receiver and was connected to the Dynamixel Smart Servo Shield in order to be able to control the AX-12A Smart Servo Robot Arm. The Uno was also be connected to the HC-12 433MHz transceivers in order to receive the serial monitor information being send from the transmitter. The Baud rate of the Arduino and HC-12 Transceivers have been set to 9600bps for uniform communication. The HC-12 will be power off the Arduino power out "5 Volts" with a 1uf Capacitor in parallel to aid in signal clarity. The Smart shield will sit on top of the Arduino Uno and the USB to Dynamixel connector will be connected to one of its ports to get the required TTL Signal to drive the AX-12A Servo Motors, the power supply for the Dynamixel Servos will come from the 240V AC Adaptor that come with the robot as the Servos draw too much current for the Arduino to power them independently. The Receiver set up can be seen in the circuit diagram below.

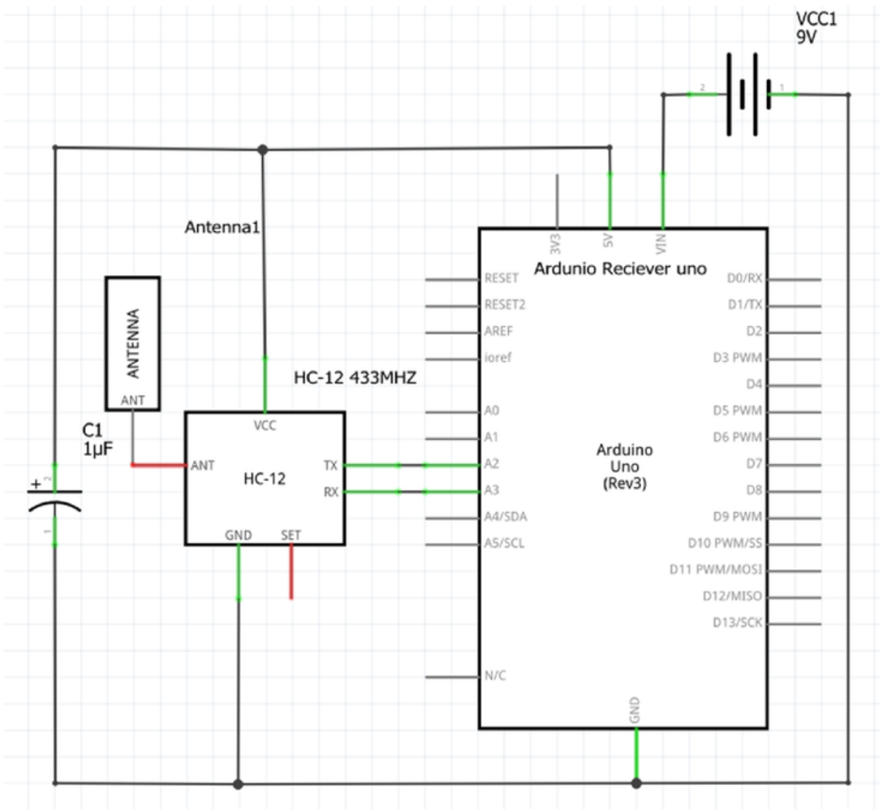


Fig 14 Receiver Circuit Configuration

The Circuit above shows the system layout minus the smart servo shield that sits on top of the Arduino which connects to the USB to Dynamixel controller in order to control the robot. The HC-12 Transceivers above receives the Data being send from the transmitting Arduino and uses it through its serial monitor in order the control the desired Servo position . The code for which can be found in Appendix B.
The Arduino itself is powered by a 9V Battery.

The above receiving circuit was changed as there was unforeseen errors that occurred in the circuit and stopped it from working according, this was a major set back for the project and appropriate action was taken to rectify the problem. The main changes were to the Arduino itself, instead of using a Arduino Uno and Dynamixel smart servo shield in order to control the robotic arm. It was changed to a Arduino Mega Pro Mini along with a SN74LS241N Line buffer in order to get the required half duplexed signal in order to control the dynamixel smart servos of which the robotic arm is made of.[14]

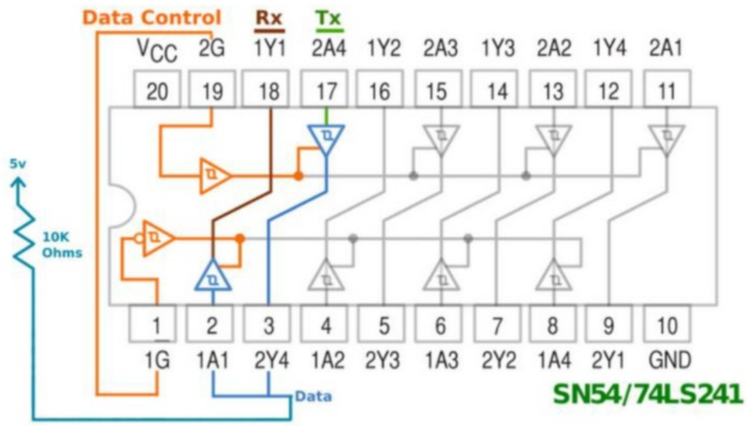


FIG 15. 74LS24 Line buffer

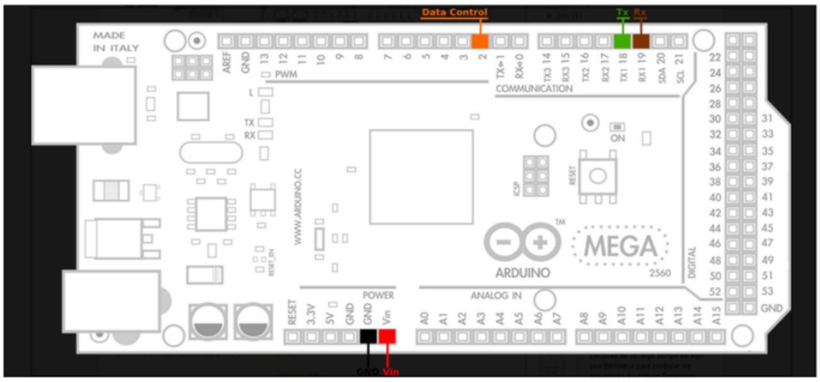


FIG 16 Arduino Mega

The circuit was changed by removing the dynamixel smart shield that gave the servos the half duplexed and replacing it with the SN74LS241N Line buffer the connection are shown FIG 14. This was needed as the command lines that were used to control the servos positions in the dynamixel library had to great of a delay built in that it in turn made the whole system run very slow and made the transmission extremely delayed and momentary at best. The Data (blue wire) was connected to the signal cable that ran the servo for control. The orange line was connected to PIN 4 of the Arduino Mega in order to send the desired signal to the line buffer to be changed into a half-duplexed signal. The RX pin on the line buffer was connected to the pin 19 on the Arduino mega and the TX pin on the line buffer was connecting to pin 18. This circuit therefore replaced the need for the dynamixel smart servo shield and allowed the system to run more smoothly as the serial monitors were not being heavily affected by the Dynamixel shield any more. This resulted in a lot smoother system and allowed the baud rate of both the HC-12 transceivers and both Arduino to run at a nice 57600bps which allowed for the transfer of more than enough data to make the wireless control of the whole system virtually prefect without communication lag. Although the HC-12 433Mhz transceivers have the abilty to send data at the rate of 115200bps, they were set are 57600bps after a few test were done and the result showed that too much data was getting lost when the transmission baud rate was that high.

5.3 Circuit Creation

The Circuit Creation was achieved by using a variety of skills which included: Soldering, Circuit board preparation, Flux , Wire stripping and Heat shrinking.
First all the pins were soldered on to both HC-12 Transceivers, Arduino Nano and Arduino Pro Mini as these components did not come pre-constructed. This was done by first pre-heating the soldering iron up to the desired temperature, next the pins were held in their desired place ready for soldering and both surfaced were cleaned with "flux" which is a agent that de-oxidizes the surfaces to allows the solder to stick better.

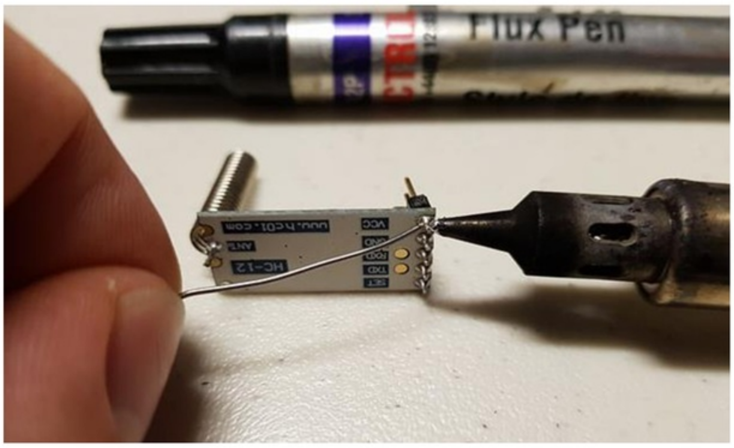


FIG17 Soldering Pins to HC-12 Transceivers

Fig17 Shows the method used in order to solder the pins onto the HC-12 Transceiver circuits, this method of the solder on one side and the heat source on the over allows for the solder to flow smoothly and evenly onto the joint. The method was repeated for both HC-12 Transceivers, Arduino Nano and Arduino MEGA Pro components.
Next all Circuits were connected with bread board as per the circuit diagrams

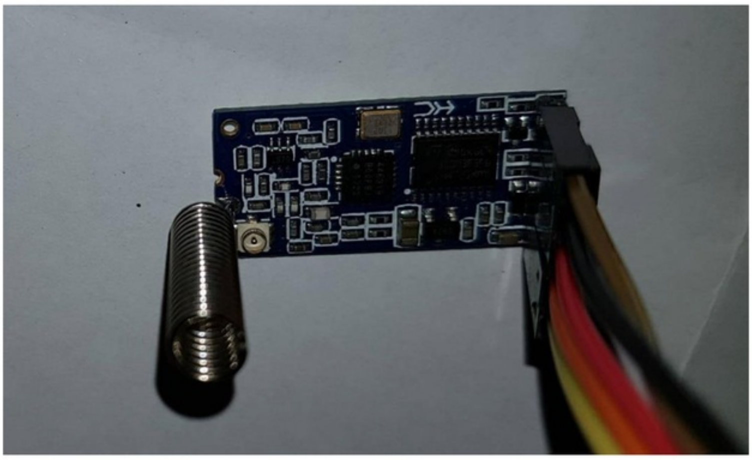


FIG18 HC-12 with pins attached

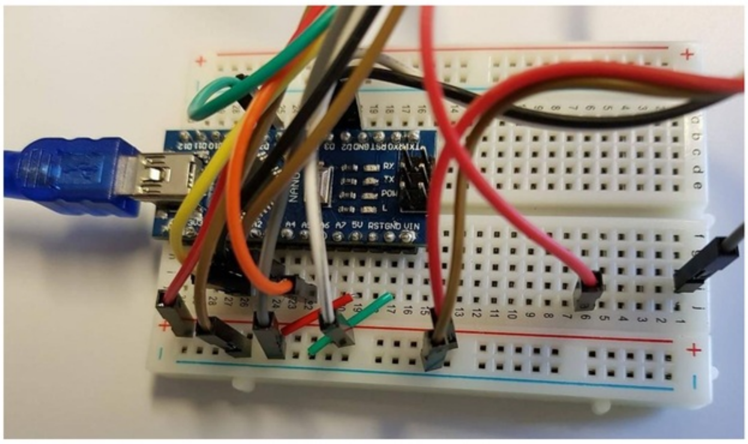


FIG19 Arduino Nano with HC-12 and Joystick wires

FIG19 Shows the pin attachments of the HC-12 transceivers and the 2x Joystick to the Arduino Nano, this circuit shown is the finalized pin layout for the transmitting circuit

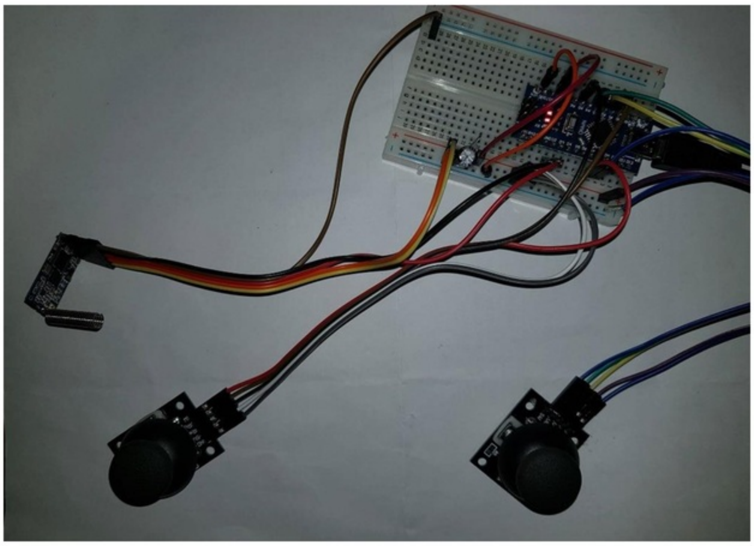


FIG20 Finalized Bread board transmitter circuit.

In FIG20 it can be seen the finalized transmitting circuit in the breadboard development stage, once all tests are commenced this circuit will be hard wired together on a development board and will be refined to have it fit in a box for easy use as a wireless controller for the robotic arm. The final controller design for hardwiring is shown below.

FIG21 Final Transmitter Controller Circuit

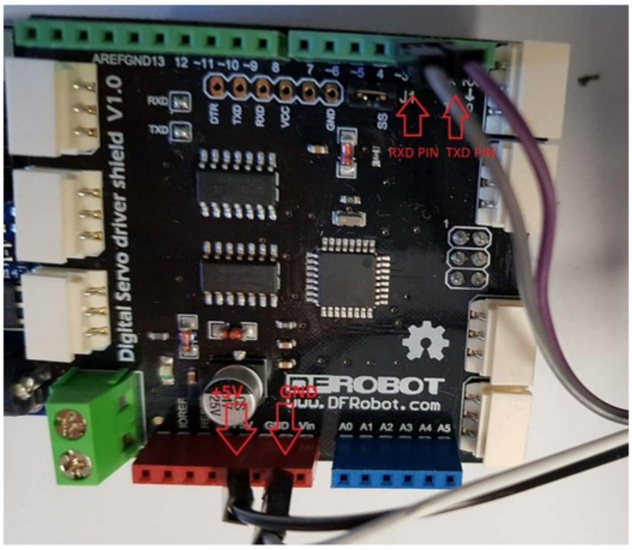


FIG 22 Original Receiver circuit

FIG22 was the Original receiving circuit for the project, it only consisted of a Arduino Uno and Smart Dynamixel Servo Shield mounted on top along with 4 wires running to power the HC-12 Transceiver. The Signal cable for the robotic arm just plugged into any of the 7 servo ports it had, this was a simple design until unforeseen reasons the system stopped working and needed to be replaced/ modified to regain the once good transmission it had. The above circuit worked well at 14ms delay as the receiver.

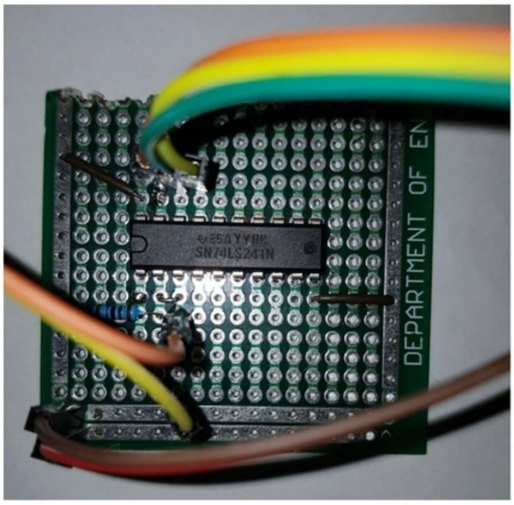


FIG23 Line Buffer Circuit to replace shield

FIG23 shows the circuit that ended up replacing the previous simple receiving circuit as it stopped worked and a solution to its unforeseen problem could not be found. This circuit is a Line Buffer and was used to split the full duplexed signal coming out of the receiving Arduino into a half-duplexed signal in order to be able for the Arduino to communicate with the Dynamixel Smart servos in the robotic arm.

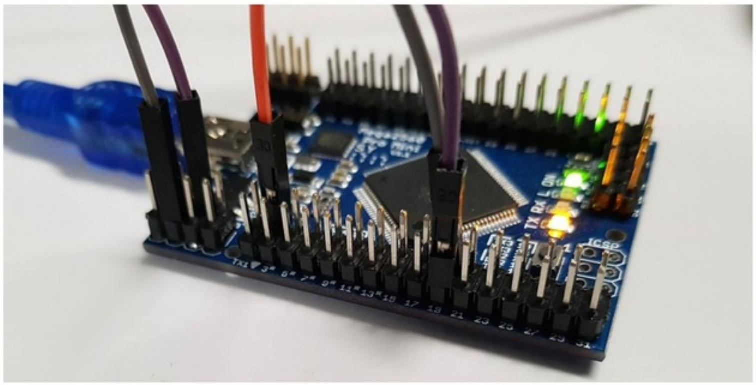


FIG24 Arduino Mega Pro Mini

Fig24 shows the Arduino Mega Pro Mini that replaced the Arduino Uno I the receiving circuit after it stopped working correctly. The Uno was replaced by a Mega and not the original Uno because in order to use the line buffer and communicate at the same time the need for multiple TXD and RXD pins were needed and as the Arduino Uno only had one set it would not be possible, so the Mega was chosen as it has 3 sets of RXD AND TXD pins.

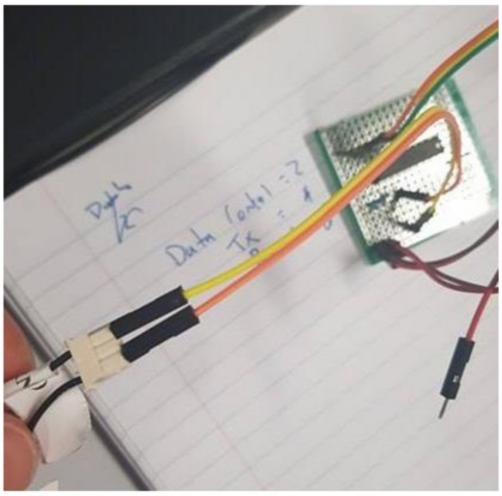


FIG25 Signal and Ground connection

FIG25 shows the connections made from the line buffer into the daisy chain cable which was used in order to communicate with all the servos within the robotic arm.

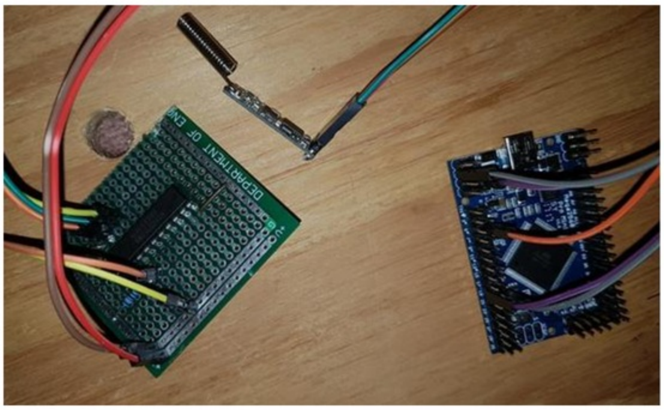


FIG26 New final receiving circuit

Fig26 shows the new final receiving circuit which consists of the Arduino Mega Pro and Line buffer instead of the original Arduino Uno and Dynamixel shield from the original design

FIG27 FSR on end effector

FIG27 shows the addition of a FSR on the robotic arms end effector in order to gain feedback once the gripper starts to put pressure on an object in order to grip it. This was added to gain functionality from the system as a whole and allowed the data to be sent back simultaneously in order for the controller to know how much newton force pressure they were applying to a certain object when they were trying to pick it up.

FIG28 Feedback Circuit

FIG28 shows the feedback circuit that was connected to the transmitting Arduino Nano In order to display the feedback data being sent back from the robotic arm. As can be send on the LCD display is prints out live data of the Nm of force the end gripper is imparting on a specific object.

FIG29 Display of Gripper Force

Fig29 Demonstrates what the user sees when they are gripping an object with the end effector of the robotic arm. This data is streamed back from the recovering circuit almost simultaneously as the robot arm is sent position commands.

FIG30 Testing FSR

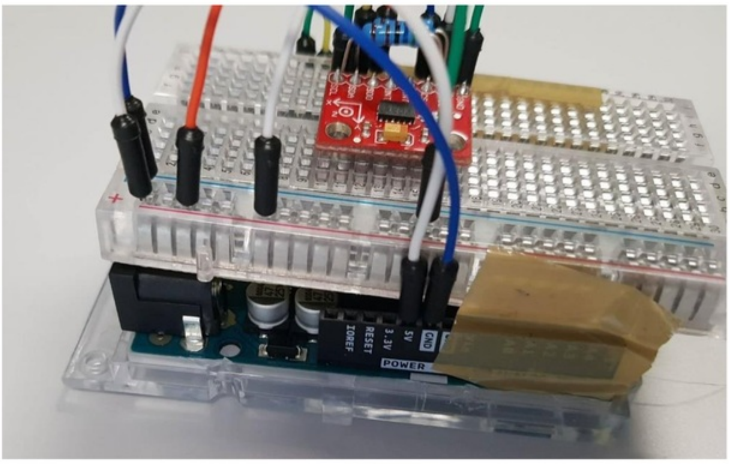


FIG31 Accelerometer test circuit

FIG31 depicts the Arduino Uno and 3-Axis Accelerometer which was used to gain data from the robotic arm while it was being designed and refined for smooth reliably operation.

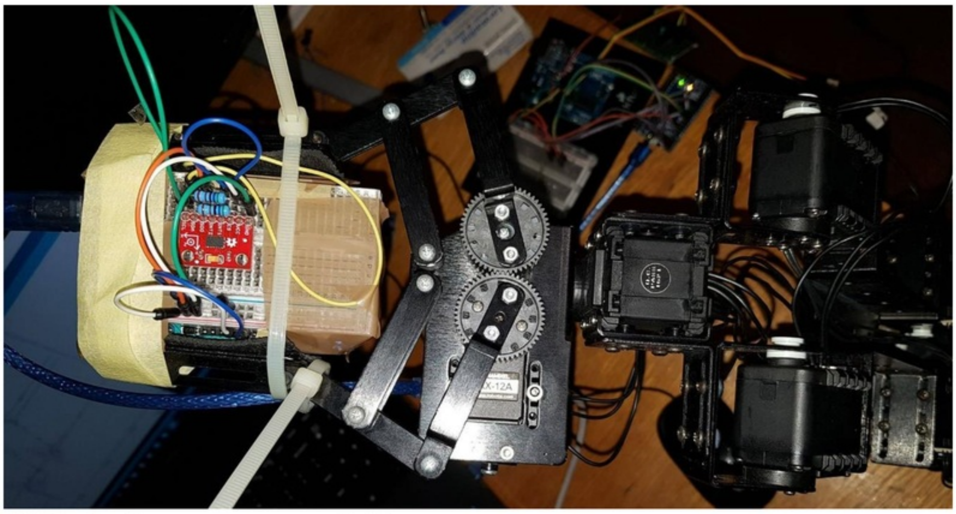


FIG32 Accelerometer test on end effector

FIG32 shows how the accelerometer circuit was attached to the robot's end effector in order to gain data on its operation and how it reacted to certain commands.

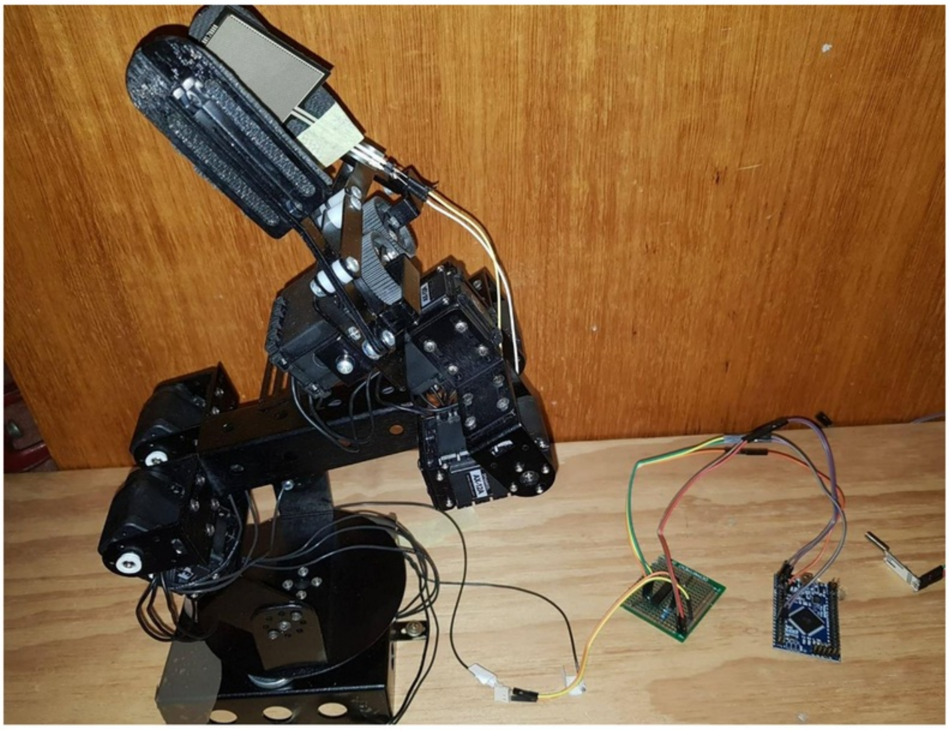


FIG33 Final 2.0 Setup of Receiving circuit

FIG34 Loctite bolts

FIG34 can be seen that the need for Loctite was needed as during the testing phase with the Accelerometer as joint 2&3 along with joint 4&5 were coming loose as a result of the movements during testing, this was needed to be rectified as it was sending wrong data to the Accelerometer as there was an unacceptable amount of play in the system.

5.4 Project Design Testing

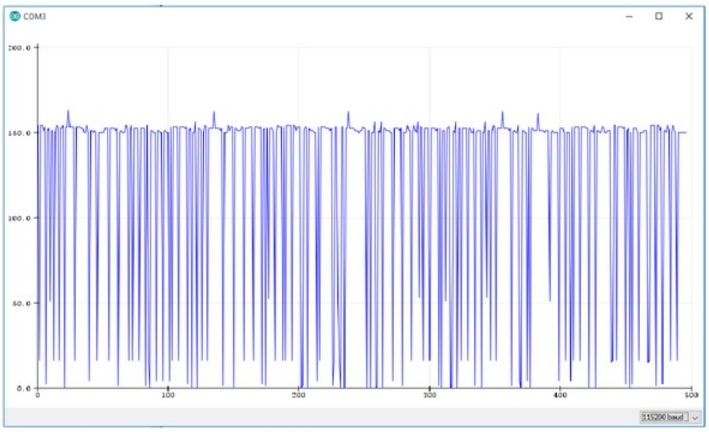


FIG 35. Test Data at 115200bps At 50MS Delay

In FIG35 it can be seen that at 115200 baud rate along with a 50ms delay the transmission of just the repeated number of 151 being send has a lot of interference. This great deviation from the numbers sent and those received was far to high and if this baud rate was used it would result in the circuit receiving lots of interference.

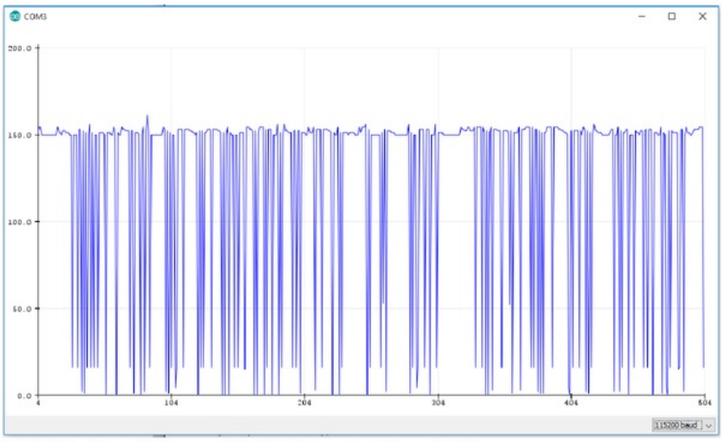


FIG 36. Test Data at 115200bps At 20ms Delay

In Fig36 it can be seen that at 115200 baud rate along with a delay of 20ms in transmission that the correct numbers been received are no better than that of the 50ms test above. These transmissions test proved that running the baud rate at 115200bps is just too fast to be able to reliably send and receive data with the HC-12 transceivers when paired with Arduinos.

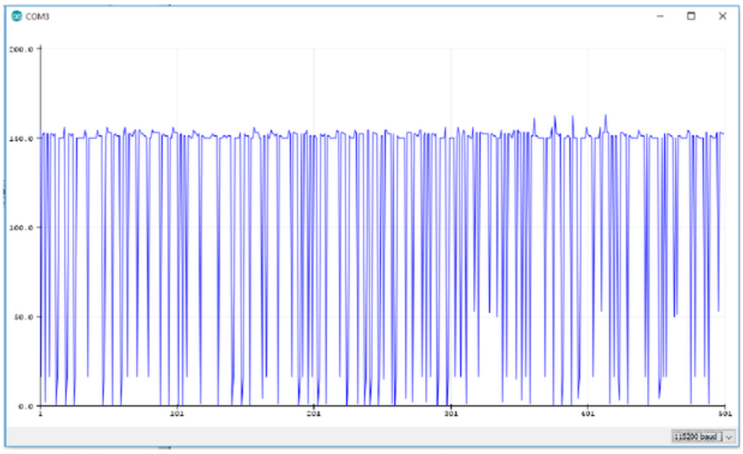


FIG 37. Test Data at 115200bps At 10ms Delay

In Fig37 it can be seen that at 115200 baud rate along with a delay of 10ms in transmission is also no good at reliable sending data when paired with Arduinos.

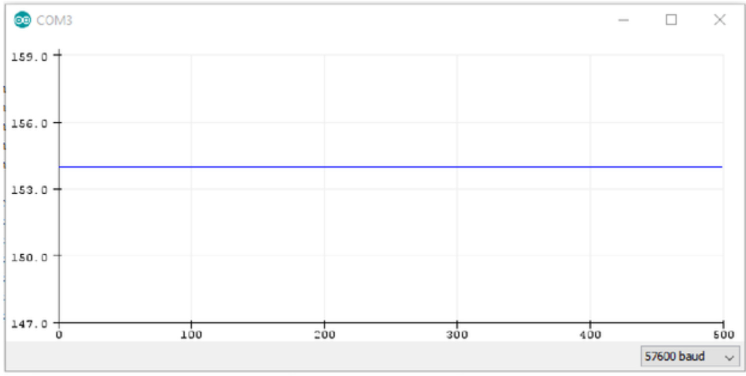


FIG 38. Test Data at 57600bps At 50ms Delay

In FIG 38 it can be seen that at 57600bps and delay 50ms that the HC-12 Transceivers can reliable send and receive data at a rate of 20Hz as 50ms calculates out too.The previous test was repeated at 20ms,10ms,5ms and 2ms with the same result, all the data was safely received. The decision was made to go with the 2ms delay as that would allow the most amount of data to be send and received which would lead to smooth system operation and faster refreshing of user defined positions so the robotic arm itself could react faster. As a delay of 2ms is 500Hz this was well high enough refresh rate for the overall system to refresh at and lead to very smooth and reliable operation. While having the transceivers originally set for 2ms was good for transmission of just 1 number it had to be reduced to 4ms in order to have reliable transmission of 4 numbers simultaneously as running through all of the code itself took time to process it caused a delay itself.

5.4 Project Design Progress

The Initial Proof of concept started by turning a led off and on by a very basic 433MHz sender and 433Mhz receiver, problem was they were not power enough to send the amount of information needed to control a 5 DOF Robotic Arm. The Proof of design concept also lacked the ability to send data back from the receivers this was a disadvantage as the force feedback feature that was planned for the robotic arms gripper would not have worked. This problem was solved by the use of HC-12 Transceivers which have a range of 1.8km and the ability to send and receive data at the same time while being power by the 5V available on the Arduino device. Along with the use of joysticks which were connected to the analogue in pins of the transmitting circuit they added the ability to send the joysticks positions wirelessly to the receiving circuit which takes those values from its serial monitor and use them to controller the AX-12A Smart Robotic Arm by the Smart Servo Shield connected on top of the Arduino Uno on the receiving side circuit.

As now the connection between the 2 wireless circuits works with a high degree of accuracy the ability to control 7 servo motors movements by the joystick on the transmitting circuit was accomplished from 10 metres away. To further the features of this design a momentary button was added to the remote in order to allow the robot to take user input to perform a predetermined task autonomously at the push of a button. Another feature that was added consisted of using a FSR "Force Sensitive resistor" in order to calculate how much force the gripper on the end effector of the arm is applying to the object it is picking up and sending the data back to the controlling circuit and printing out the Nm force on a LCD display that the user can view while controlling the arm.

The challenge of sending data both ways between the transceivers proved to be challenging as the transceivers them self could do both operation simultaneously the Arduino program commands could not , this resulted in a delay in the operation when the feedback data was being send back so a for loop was added so that only when the gripper is closing the data of the FSR on the end effector is been sent back so that it did no hinder the operation of the all joints all of the time only when the gripper was being used.

5.5 Full System testing

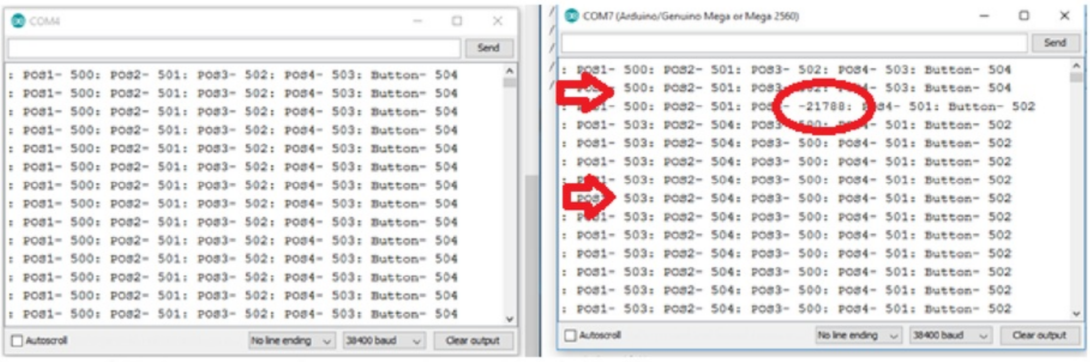


FIG39 Data streams from transmitter and receiver

FIG39 shows the 2 data streams where most of the testing was done, the left stream (COM4) shows the number being calculated in the transmitting circuit and being serial. Printed to the monitor while also being transmitted to the receiver circuit. No problems raised from the transmitting circuit as the number being calculated are only within that Arduino and are not being received so this results in no errors on the left hand side(transmitting circuit).

On the right side (receiving circuit) is a different story, as all the data being printed has been sent from the left data stream (transmitting circuit). Since this data has been sent there is always going to be bits of information that not get transmitted correctly, these errors have been highly in red to show the errors and effect they have on the streams of data. As can be seen in the first 3 lines in each stream the data in prefect and uniform, but when the receiving circuit gets to the 4th line it experiences some interference which results in the data being thrown off and are no longer in line. This was a major problem as then each servo would not receive the correct data in order to move to that desired position so this problem needed to be rectified. This problem was fixed by adjusting the baud rate from 38600bps to 57600bps along with changing the delay in the transmitting circuit to 12ms and 4ms for the receiving circuit. As with transferring data the importance of both Arduino circuit and programs running at the same speed is extremely important in order to have high reliability.

Chapter 6

Results

The next stage of testing that was completed after the system was working theoretically was to test the systems practically. This was done by connecting the Dynamixel Smart robotic arm to the receiving circuit and to send certain commands to each joint to test the overall system performance.

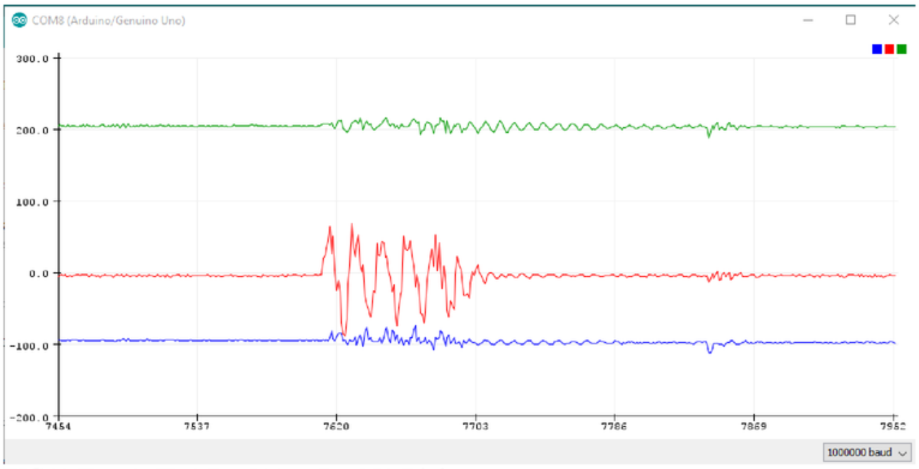


FIG40 Test 1 ID 1 Pos500 to Pos250 Right turn

FIG40 shows the first of a series of tests that was used to refine the systems movement. Each joint was tested individually through a range of movements multiple times and the averages were graphed. Test 1 was the movement of the first servo "ID 1" which is responsible for rotating the robotic arm about it base plate.

The graphs are showing the changes in the X-direction "Blue" ,Y-Direction "Red" and the Z-Direction "Green". The full movement range of this joint is 300 degrees when unrestricted by the Arduino code. The test was formed from imputed the data from the joystick electronically instead of their physical movements, this lead to the program seeing the joystick move like they usually do once they have an input and will react the exact same way, this was done to ensure all testing of all joints was linear and can be compared against each other.

In the Arduino code the joint movements are from 0 to 1023 which allows the dynamixel servo to move from 0 degrees to 300 degrees. Therefore, this movement from 500 to 250 is equal to moving from 122.18 degrees to 73.3 degrees in the clockwise direction.

The movement can be seen in FIG## , although this movement seems smooth to the naked eye , the accelerometer data reads a different story, as the data is sent and refreshed many times a second the total move from position 122.18 degrees to 73.3 degrees is made up of lots of small movements happening faster than the eye can process this is why this kind of testing must be carried out by an accelerometer as it is getting data refreshed and printed out at 1Mbps which is extremely high when compared to the 57600bps that the system runs out.

Therefore, having the accelerometer run faster than the system allows the system to be refined to a higher degree than what is possible by just human observation. These tests were repeated for all the joints in order to smooth out there operation along with a few different movements in each direction as the gravity that acts on the arm moving down effects its movement differently to when its in the opposite direction moving up against gravity.

This in turn can be seem in a number of the tests that when the joint was moving against gravity its settling time was faster than when the joint was moving in the same direction as gravity.

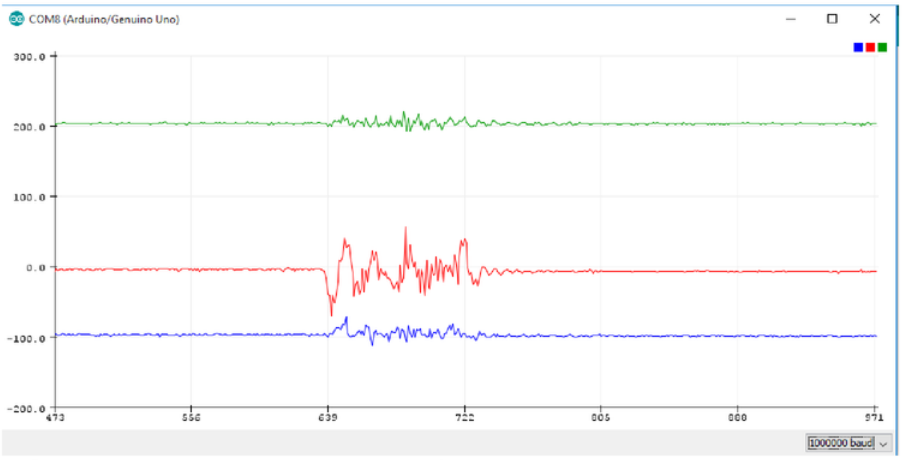


FIG41 Test 2 ID 1 Pos500 to Pos750 Left turn

FIG41 shows the exact same movement in the opposite direction as Test 1 , it can be seen that the movement and settling time is not as smooth as that of Test 1 this is that there is very small discrepancies in the motors they result in them favouring 1 direction compared to the other.

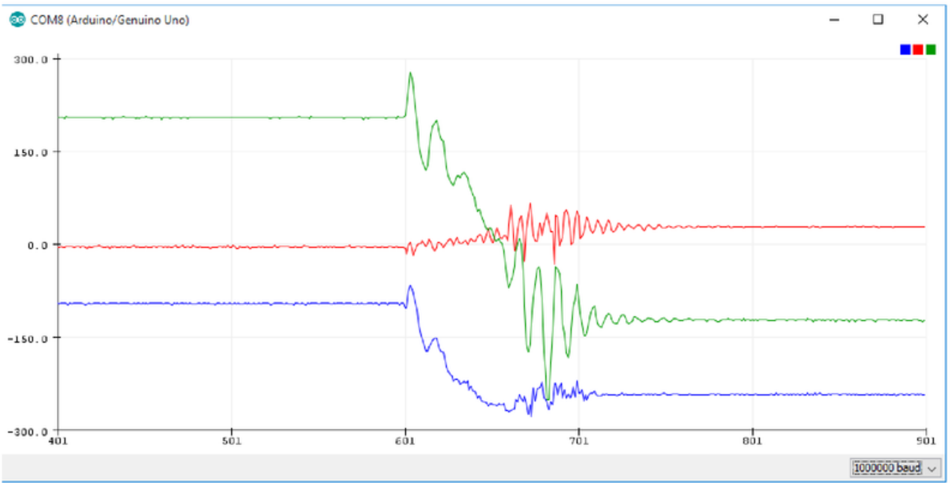


FIG42 Test 3 ID 2&3 Pos500 to Pos750 "Up"

FIG42 shows that the movement up from 122.18 degrees to 183.28 degrees, as gravity was against this movement the settle time is ok in the Y-Direction as the motor is fighting gravity to move.

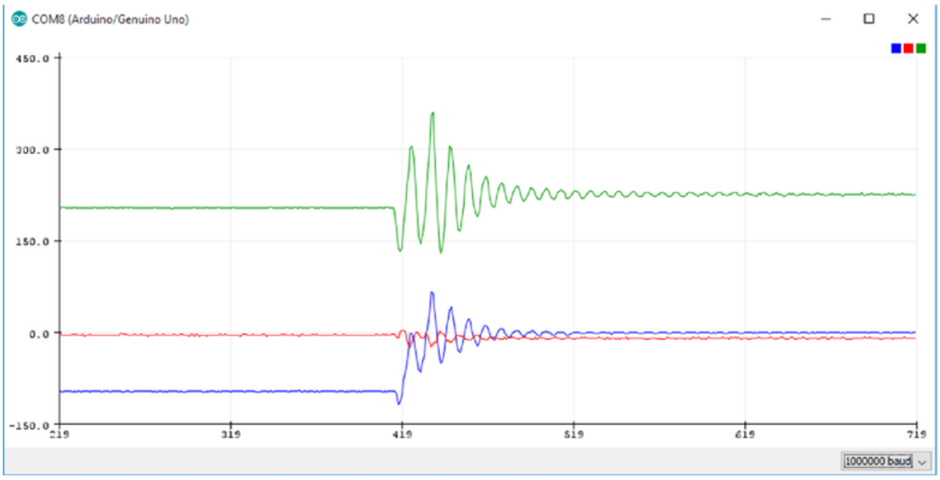


FIG43 Test 4 ID 2&3 Pos500 to Pos250 "Down"

In test 4 as the movement of the arm was with gravity, this resulted in an extremely nice settling time which was very linear. This is what all joint were fine tuned to resemble. This was done by changing the values added in certain joystick direction along with the delay time in those particular direction also.

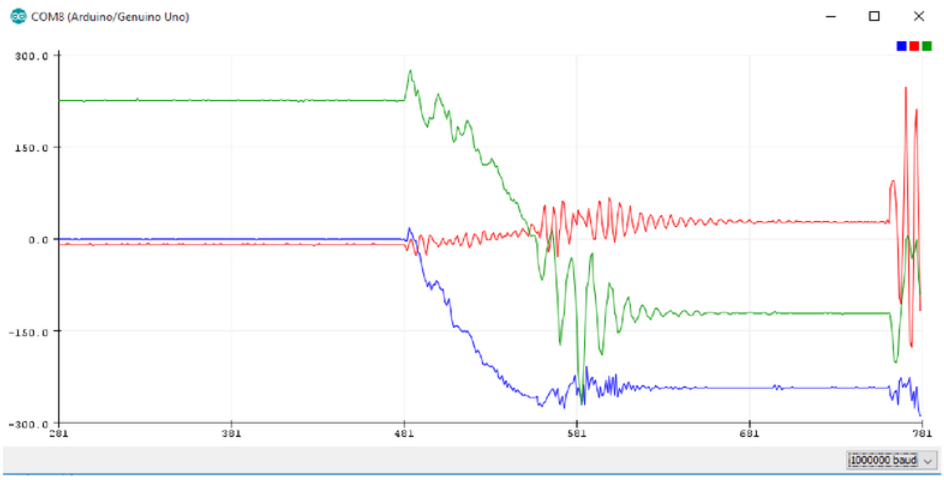


FIG44 Test 5 ID 2&3 Pos250 to Pos750 "Up"

Test 5 was similar to test 3 only difference was the arm moved through a wider angle "73.3 to 183.28 Degrees". The movements are quite linear although the settling time is a little higher as the robot gained a little more speed in this test.

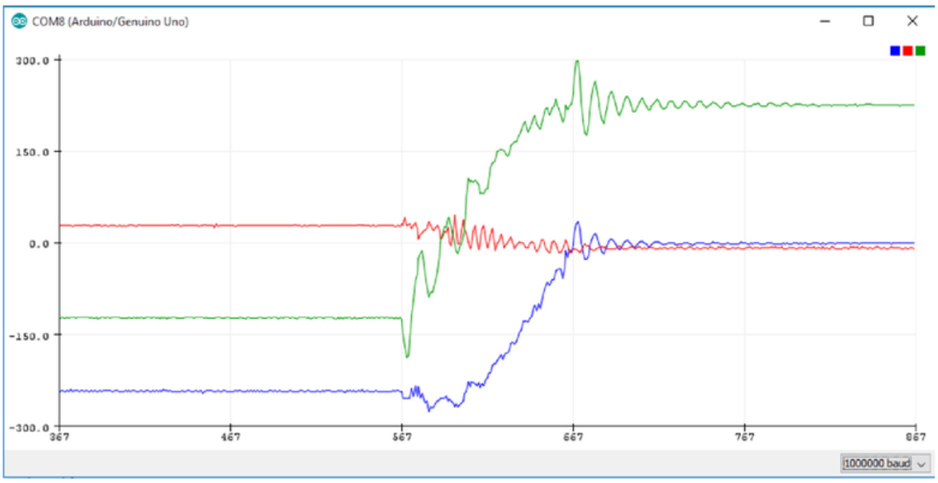


FIG45 Test 6 ID 2&3 Pos750 to Pos250 "Down"

Test 6 was quite similar to test 4 as the arm just moved through a slightly wider range of positions. Just like Test 4 the range of motion is very smooth along with a extremely nice settling time.

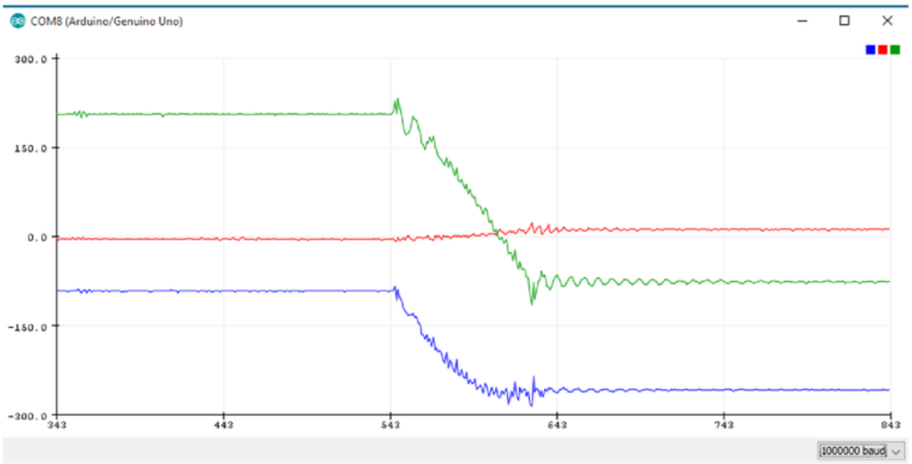


FIG46 Test 7 ID 4&5 Pos500 to Pos250 "Up"

Test 7 shows similarity to Test 2 as both times the accelerometer are moving in the same direction only difference is there is a lot less load on joint "ID4&5" compared to that of joint "ID2&3" this have a major effect on the smoothness of the movement along with their settling times, as this joint has less weight acting on it. The overall motion is a lot smooth when compared to that of Id 2&3 while both joint have 2 motors this joint has less weight acting on it and therefore can do the controlled task we greater ease. This is the reason for its such nice movement and settling time seen above.

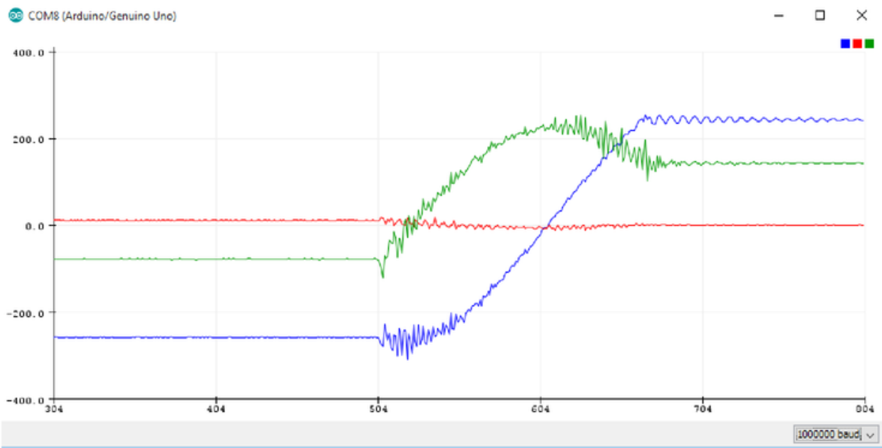


FIG47 Test 8 ID 4&5 Pos250 to Pos750 “Down”

Test 8 can be seen to be one of the best results as it shows the motion of the 3rd last effector moving with ease. Nothing needed changing in the code to improve its function as its movement and settle time was great to start with. This test is what most over motors were based off when fine turning them for overall operation.

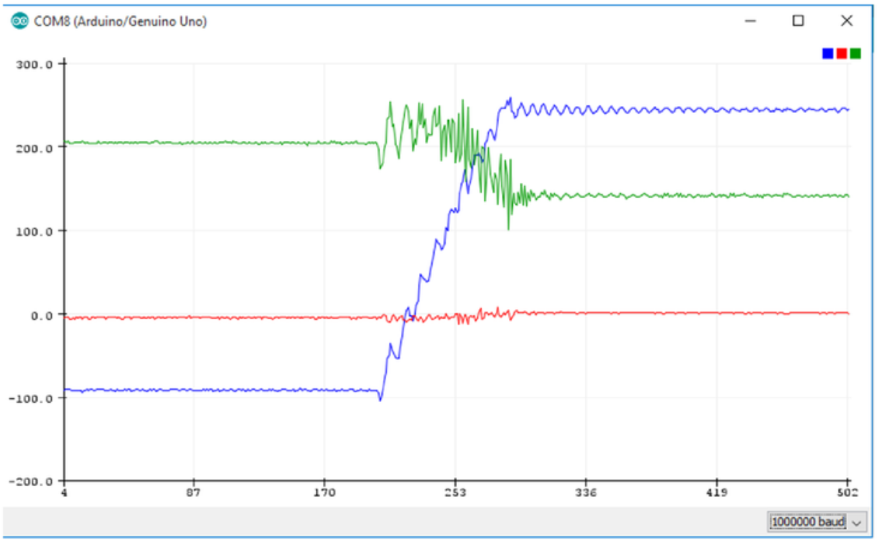


FIG48 Test 9 ID 4&5 Pos500 to Pos250 “Down”

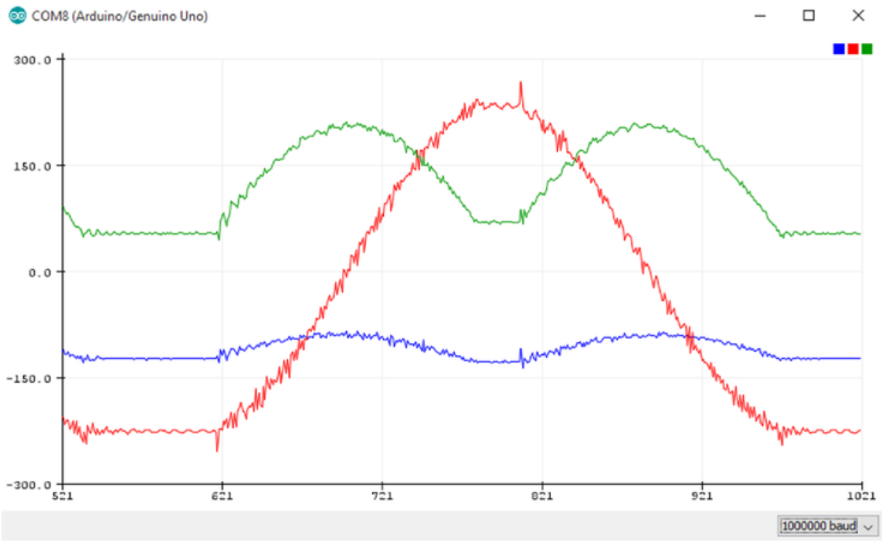


FIG49 Test 10 ID 6 Pos750 to Pos250 to Pos750 "Rotate"

As can be seen from test 10 this was the smoothest result so far, this was mainly due to this joint being the rotation "wrist" joint in the robotic arm. This meant that this joint had the least amount of weight to move of any of the joints and could take its instructions and move according the best. When analysing these results, it can be seen that the system could be improved by having a lighter construction along with stronger motors that would have more torque which would allow for less of effects on the servos while they were moving under the weight of the robot.

Chapter 7

Discussion and Conclusion

7.1 Discussion

The Thesis project undertaken has been to control a smart Dynamixel robotic arm wirelessly to be able to accomplish the designated task of picking up and dropping a tiny object autonomously. The task has been done multiple times with a high degree of accuracy. By using the AX12A smart robotic arm along with 1 Arduino Nano, 1 Arduino Mega, a SN74LS241 Line buffer and 2x HC-12 433MHZ transceivers the stated project was able to be delivered.

The main deliverables were able to be obtained by the wireless communication transceivers which allowed the transmission of data between the 2 separate circuits serial monitors, when couple with a SN74LS241 Line buffer which allowed the Arduino coding language to be transformed into the half duplexed signal needed for the control of the AX-12A smart servos this was made possible . The sending circuit was able to take analogue values from the joysticks connected for control and then send those number wirelessly to the receiving side circuit which then allowed the SN74LS241 Llne buffer to speak to the robotic arm to change its position to the desired one. The circuit prototyping aided with the systems final design as some certain commands in Arduino would allow the desired task to be achieved but not with a high degree of reliability, through further research into how certain command lines in Arduino work, the system was able to be refined to get a smooth communication and reliable control of the robotic arm. A simulation of this system was thought out as it may aid in the scope of the overall project.

The project was a great challenge in terms of research and problem solving abilty, the first of many hurdles including determining the right wireless communication device in order to get the desired results, the second and most challenging problem that occurred is that the original system design work perfectly the whole time up until 1 week before its completion the whole system failed and a solution was needed to be found, the cause was not fully understood at the time but was pinned down to the specific commands lines used in the dynamixel smart servo library that causes to larger of a delay that it started to mess with the commutation of the 2 circuits. This problem only occurred when the computer being worked on crashed and had to have a hard reset , even though all data was recovered the code and system never worked the same since. This was narrowed down to a command that used to work fine in Arduino which must have changed when the newer version of Arduino was installed on the device after this point in time the system stopped working, the way the certain command line used to work was most likely changed during the update. This problems was worked on for 4 days straight, 10hrs each day and could not be solved so an alternative system had to be design and created fast.

This resulted in the receiver circuit no longer using the Smart Dynamixel servo shield anymore, instead it was replaced with an Arduino Mega Pro and a SN74LS241 Line buffer when both these pieces of hardware were merged they created a system in which the data being received from the transmitting circuit could be gathered and sent to the Dynamixel motors at an extremely fast rate, this in term ended up working better than the original design did for the first few months. This current design had the feedback circuit also linked to it so the user can have the Nm of force they are putting on object when they pick it up sent back to them and displayed on a LCD on the controller itself, this feedback featured added great functionally to the system and went on to show that the more sensors you can have feedback data to the user the more refined the robotic arm could be made.

The limitations in this project as a whole was from the wireless communication as there is a limit to how much data you can send and receive at the same time. This is heavily dependant on the wireless device being used, as the HC-12 transceivers used for this project was plenty powerful enough to send all the constantly updated servo positions of 7 servos and also sending back data from a FSR at the same time. If more functions were to be added to the robotic arm maybe a more powerful transceiver could be researched. From the results obtained the previous question in thought was able to be answered and refined. The test data showed that the system was more than capable of reaching its desired needs along with the abilty or refinement in certain effectors of the robot. It could be seen as a valid project and provided great insight into the world of wireless communication.

This project shows what impact robotic technology has as a whole and how it can be used to benefit manufacturing and construction business. The wireless section of this project was important as the larger the amount of data that can be sent every second allows the robot to have the most up to date control by minimising lag. The current configuration can be improved upon by adding more feedback functions which would allow the user to receive sensor data from the smart robotic arm being controlled to greater the systems overall functionality and design.

7.2 Conclusion

To Conclude, the ability to control an AX-12A Smart robotic arm through wireless means in order to pick up and drop a tiny object autonomously has been made possible by the use of 1 Arduino Nano, 1 Arduino Mega Pro Mini, SN74LS241 Line buffer and 2x HC-12 433Mhz Transceivers. This has been done reputably with high amounts of accuracy. This was refined and made possible by the research in circuit creation and testing that was done on both circuits. Though there were problems along the way they were resolved by engineering practices and documented for future reference. A project schedule with a full work breakdown, including suggested as well as possible processes, were covered in this report. The detailed description of testing procedures were presented and the results interpreted to aid in the design. The future work and potential solutions to future problems have also been discussed. This document follows the guidelines that were specified in the abstract. It can be concluded that many tasks are capable of being done by wireless communication, the only restricting factor is the speed in which data can be sent and received reliably.

Chapter 8

Future Work

Although robotic arm type systems already exist their consistent refinement can always be improved and researched in further detail. Some more of the features planned to be added to the design to increase useability which would include more FSR to gain feed back to the controller along with a LCD display in which the user can view to gain even more feedback data about the robotic arms position and force its applying to a given object. The use of a range of sensors that can be used for obstacle avoidance which could include but is not limited to ultrasonic distance sensor, ir sensors along with microwave sensor, laser sensors. A push button could also be implemented in order to start a predetermined autonomously sequence that the robotic arm can follow. Further refinement in the amount of data that specific wireless transceivers can send along with further research into different types such as Bluetooth, WIFI or 2.4Ghz RF.

Chapter 9

Abbreviations

LCD	Liquid Crystal Display
FSR	Force Sensitive Display
PID	Proportional Integral Derivative
DOF	Degrees of Freedom
LED	Light Emitting Diode
NC	Numerical Control
RF	Radio Frequency
POS	Position

Appendix A

Project Plan and Gantt Chart

A.1 Overview

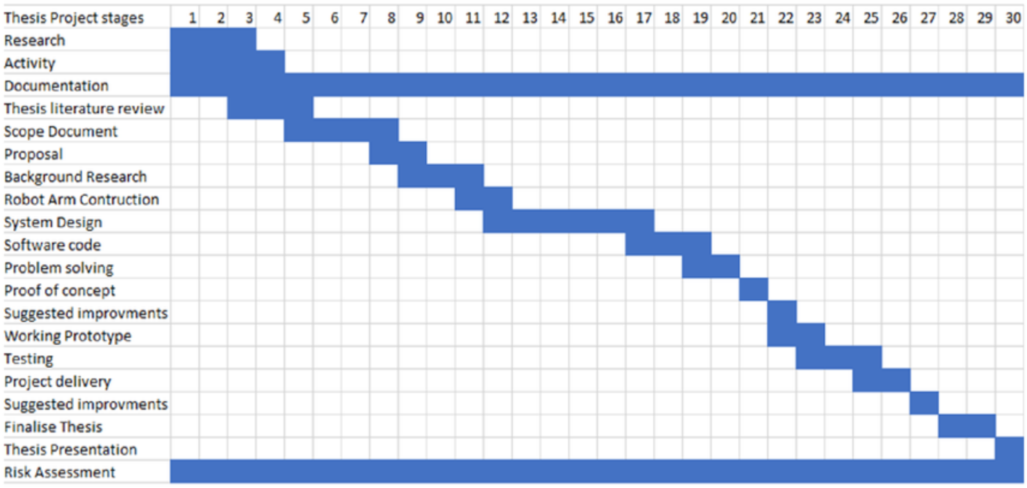
The following Appendix consists of the tables and graphs that were used in order to plan out the project time line and deliverable

A.2 Project Plan

ENGG411 Research Thesis Project	4/09/17	
Tasks		
Name	Start Date	Finish Date
Research	27/02/2017	11/11/2017
Activity	27/02/2017	11/11/2017
Documentation	27/02/2017	11/11/2017
Thesis Literature review	27/02/2017	24/03/2017
Scope Document	30/03/2017	21/04/2017
Proposal	1/03/2017	19/03/2017
Background Research	19/03/2017	11/11/2017
Robot Arm Construction	31/07/2017	4/08/2017
System Design	07/08/2017	11/08/2017
Software Code	13/08/2017	20/08/2017
Problem solving	27/02/2017	11/11/2017
Proof of concept	20/08/2017	28/08/2017
Progress Report	21/07/2017	04/09/2017
Working prototype	10/09/2017	15/09/2017
Testing	15/09/2017	21/10/2017
Project Delivery	21/07/2017	Exam Time
Finalise Thesis	21/07/2017	6/11/2017
Abstract	21/07/2017	11/11/2017
Log Book	Week 1	Week 14
Thesis Poster	21/07/2017	11/11/17
Thesis Presentation	21/07/2017	Exam Time
Risk Assessment	21/07/2017	Exam Time

A.3 Gantt Chart

ENGG411 Research Thesis Project



A.4 Consultation Forms

Consultation Meetings Attendance Form

Week	Date	Comments (if applicable)	Student's Signature	Supervisor's Signature
1	1/8/17	First Meeting	J.B. [illegible]	[illegible]
1	4/8/17	Meeting of last day wk.	J.B. [illegible]	[illegible]
2	8/8/17	Proposed Potential Project Plan	J.B. [illegible]	[illegible]
2	11/8/17	updated progress so far.	J.B. [illegible]	[illegible]
3	17/8/17	Project basics Working	J.B. [illegible]	[illegible]
4	22/8/17	Project Update	J.B. [illegible]	[illegible]
4	25/8/17	Update	J.B. [illegible]	[illegible]
5	29/8/17	Basic System Working	J.B. [illegible]	[illegible]
6	4/8/17	Update	J.B. [illegible]	[illegible]
6	7/9/17	Coded base for pos movem	J.B. [illegible]	[illegible]
7	14/9/17	Rough Movement of 2 Joints	J.B. [illegible]	[illegible]
8	21/9/17	Update	J.B. [illegible]	[illegible]

Consultation Meetings Attendance Form

Week	Date	Comments (if applicable)	Student's Signature	Supervisor's Signature
9	25/9/17	Update	[illegible]	[illegible]
10	2/9/17	Finilizing Design.	[illegible]	[illegible]
10	6/9/17	Demo Prototype	[illegible]	[illegible]
11	2/10	Project Update	[illegible]	[illegible]
12	10/10	Update	[illegible]	[illegible]
13	17/10	Update	[illegible]	[illegible]

Appendix B

Arduino Code

B.1 Transmitter Code

```
//HC-12 Transmitting code
const int lbuttonPin = 8; // digital pin connected to switch output
const int rbuttonPin = 9; // digital pin connected to switch output
int lbuttonState = 0;
int rbuttonState = 0;
const int LX_pin = A5; // analog pin connected to X output
const int LY_pin = A1; // analog pin connected to Y output
const int RX_pin = A2; // analog pin connected to X output
const int RY_pin = A3; // analog pin connected to Y output
int LX = analogRead(LX_pin);
int LY = analogRead(LY_pin);
int RX = analogRead(RX_pin);
int RY = analogRead(RY_pin);
int POS1 = 500;
int POS2 = 501;
int POS3 = 502;
int POS4 = 503;
int POS5 = 504;
int i=0;

#include <SoftwareSerial.h>
      SoftwareSerial mySerial(2, 3); //TX, RX

void setup() {
  pinMode(lbuttonPin, INPUT);
  pinMode(rbuttonPin, INPUT);
  digitalWrite(lbuttonPin, HIGH);
  digitalWrite(rbuttonPin, HIGH);
```

```
  mySerial.begin(57600);
  Serial.begin(57600);
    delay(1000);
}
void loop() {

  LX =analogRead(LX_pin);
  LY =analogRead(LY_pin);
  RX =analogRead(RX_pin);
  RY =analogRead(RY_pin);
   i++;
  if( i>200 && i< 799){                    //Code for tetsing
    if (POS1 < 750) {
      POS1 = POS1 + 2;
    }
  }
if( i>800){
    if (POS1 > 0) {
      POS1 = POS1 - 2;
    }
  }

  if (LX >= 600) {          //Servo 1 Movement
    if (POS1 < 1000) {
      POS1 = POS1 + 2;
    }
  } else if (LX <= 400) {
    if (POS1 > 0) {
      POS1 = POS1 - 2;
    }
  }
  if (LY >= 600) {          //Servo 2&3 Movement
    if (POS2 < 1000) {
      POS2 = POS2 + 2;
```

```
  }
} else if (LY <= 400) {
  if (POS2 > 0) {
    POS2 = POS2 - 2;
  }
}
if (RX >= 600) {          //Servo 4&5 Movement
  if (POS3 > 0) {
    POS3 = POS3 - 2;
  }
} else if (RX <= 400) {
  if (POS3 < 1000) {
    POS3 = POS3 + 2;
  }
}
if (RY >= 600) {          //Servo 6 Movement
  if (POS4 > 0) {
    POS4 = POS4 - 2 ;
  }
} else if (RY <= 400) {
  if (POS4 < 1000) {
    POS4 = POS4 + 2;
  }
}
lbuttonState = digitalRead(lbuttonPin);
if (lbuttonState ==  LOW) {
  if (POS5 < 1000) {
    POS5 = POS5 + 1  ;
  }
}
rbuttonState = digitalRead(rbuttonPin);
if (rbuttonState ==  LOW) {
  if (POS5 > 90) {
    POS5 = POS5 - 1  ;
```

```
    }
  }

// if (POS5<145){
// int force = mySerial.parseInt();
//Serial.print(": force- ");
// Serial.print(force);
// }

  mySerial.println(POS2);
  mySerial.println(POS3);
  mySerial.println(POS4);
  mySerial.println(POS5);
    mySerial.println(POS1);



  Serial.print(": POS1- ");
  Serial.print(POS1);
  Serial.print(": POS2- ");
  Serial.print(POS2);
  Serial.print(": POS3- ");
  Serial.print(POS3);
  Serial.print(": POS4- ");
  Serial.print(POS4);
  Serial.print(": Button- ");
  Serial.println(POS5);

    delay(14);//delay little for better serial communication
//mySerial.flush();            //clear the serial buffer for unwanted inputs
}
```

B.2 Receiver Code

```
#include <DynamixelSerial1.h>
#include <SoftwareSerial.h>
#include <SPI.h>
#include "pins_arduino.h"
SoftwareSerial mySerial(10, 11); // RX, TX
void setup() {
  digitalWrite(SS, HIGH);
  SPI.begin ();
  SPI.setClockDivider(SPI_CLOCK_DIV8);
  Dynamixel.begin(1000000, 4); // Inicialize the servo at 1Mbps and Pin Control 2
  mySerial.begin(57600);
  Serial.begin(57600);
  delay(1000);
}
void loop() {

  int POS1 = mySerial.parseInt();
  int POS2 = mySerial.parseInt();
  int POS3 = mySerial.parseInt();
  int POS4 = mySerial.parseInt();
  int POS5 = mySerial.parseInt();

  Dynamixel.moveSpeed(1, POS2, 60); // Move the Servo radomly from 200 to 800
  Dynamixel.moveSpeed(2, POS1, 60); // Move the Servo radomly from 200 to 800
  Dynamixel.moveSpeed(3, POS1, 60); // Move the Servo radomly from 200 to 800
  Dynamixel.moveSpeed(4, POS3, 60); // Move the Servo radomly from 200 to 800
  Dynamixel.moveSpeed(5, POS3, 60); // Move the Servo radomly from 200 to 800
  Dynamixel.moveSpeed(6, POS4, 50); // Move the Servo radomly from 200 to 800
  Dynamixel.moveSpeed(7, POS5, 50); // Move the Servo radomly from 200 to 800

  Serial.print(": POS1- ");
  Serial.print(POS1);
```

```
 Serial.print(": POS2- ");
 Serial.print(POS2);
 Serial.print(": POS3- ");
 Serial.print(POS3);
 Serial.print(": POS4- ");
 Serial.print(POS4);
 Serial.print(": Button- ");
 Serial.println(POS5);

// mySerial.flush();            //clear the serial buffer for unwanted input
 mySerial.flush();
  delay(4);
```

B.3 HC-12 Baud rate set Code

```
include <SoftwareSerial.h>

SoftwareSerial mySerial(2, 3); // RX, TX

void setup() {
  Serial.begin(9600);
  Serial.println("Enter AT commands:");
  mySerial.begin(9600);
}

void loop(){
  if (mySerial.available()){
    Serial.write(mySerial.read());
  }
  if (Serial.available()){
    mySerial.write(Serial.read());
  }
}
```

B.4 FSR LCD TEST CODE

```
#include <LiquidCrystal.h>
const int FSR_PIN = A0;
const float VCC = 4.98;
const float R_DIV = 3300.0;
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  lcd.begin(16, 2);
  lcd.print("Clamp Nm");  // Print a message to the LCD.
 Serial.begin(38400);
  pinMode(FSR_PIN, INPUT);
}

void loop() {
  //lcd.setCursor(0, 1); // set the cursor to column 0, line 1   // (note: line 1 is the
second row, since counting begins with 0):
 // lcd.print(force);
int fsrADC = analogRead(FSR_PIN);

  if (fsrADC != 0) // If the analog reading is non-zero
  {
    // Use ADC reading to calculate voltage:
    float fsrV = fsrADC * VCC / 1023.0;

    float fsrR = R_DIV * (VCC / fsrV - 1.0);
    //Serial.println("Resistance: " + String(fsrR) + " ohms");
    // Guesstimate force based on slopes in figure 3 of
    // FSR datasheet:
    float force;
    float fsrG = 1.0 / fsrR; // Calculate conductance
    // Break parabolic curve down into two linear slopes:
```

```
  if (fsrR <= 600)
    force = (fsrG - 0.00075) / 0.00000032639;
  else
    force =  fsrG / 0.000000642857;
    force = (force*9.81) / 1000;
  Serial.println(String(force) + "N");
  Serial.println();
 lcd.setCursor(9, 0); // set the cursor to column 0, line 1   // (note: line 1 is the second
row, since counting begins with 0):
 lcd.print(force);
lcd.setCursor(13, 0); // set the cursor to column 0, line 1   // (note: line 1 is the second
row, since counting begins with 0):
 lcd.print("N");
  delay(120);
 }
```

References

[1] En.wikipedia.org. (2017). *Industrial robot*. [online] Available at: https://en.wikipedia.org/wiki/Industrial_robot [Accessed 3 Sep. 2017].

[2] ElProCus - Electronic Projects for Engineering Students. (2017). *Pick and Place Robot Arm and Movement Controlled by Wireless RF Module*. [online] Available at: https://www.elprocus.com/pick-n-place-robot/ [Accessed 3 Sep. 2017].

[3] HowStuffWorks. (2017). *How Robots Work*. [online] Available at: http://science.howstuffworks.com/robot2.htm [Accessed 3 Sep. 2017].

[4] Anon, (2017). [online] Available at: http://www.scientiareview.org/pdfs/353.pdf [Accessed 3 Sep. 2017].

[5] En.wikipedia.org. (2017). Robotic arm. [online] Available at: https://en.wikipedia.org/wiki/Robotic_arm [Accessed 3 Sep. 2017].

[6] Qureshi, M. and Syed, R. (2017). The Impact of Robotics on Employment and Motivation of Employees in the Service Sector, with Special Reference to Health Care.

[7] Anon, (2017). [online] Available at: http://file:///C:/Users/justin/Downloads/Robotics_Safety.pdf [Accessed 3 Sep. 2017].

[8] Turek, Fred D. (June 2011). "Machine Vision Fundamentals, How to Make Robots See". NASA Tech Briefs. 35 (6): 60–62. [Accessed 4 Sep. 2017].

[9] "History of Industrial Robots" (PDF). [Accessed 4 Sep. 2017].

[10] "Singularities in six-axis vertically-articulated industrial robots". CoRo Blog [Accessed 4 Sep 2017].

[11] Www2.warwick.ac.uk. (2017). Cite a Website - Cite This For Me. [online] Available at: https://www2.warwick.ac.uk/fac/sci/physics/research/condensedmatt/imr_cdt/students/david_goodwin/teaching/mechatronics/l10_robotarms_cis009_2.pdf [Accessed 1 Nov. 2017].

[12] Savageelectronics.blogspot.com.au. (2017). *Arduino y Dynamixel AX-12*. [online] Available at: http://savageelectronics.blogspot.com.au/2011/01/arduino-y-dynamixel-ax-12.html [Accessed 2 Nov. 2017].

[13] Support.robotis.com. (2017). *AX-12/AX-12+/AX-12A*. [online] Available at: http://support.robotis.com/en/product/actuator/dynamixel/ax_series/dxl_ax_actuator.htm [Accessed 5 Nov. 2017].

[14] Www2.warwick.ac.uk. (2017). *Cite a Website - Cite This For Me*. [online] Available at: https://www2.warwick.ac.uk/fac/sci/physics/research/condensedmatt/imr_cdt/students/david_goodwin/teaching/mechatronics/l10_robotarms_cis009_2.pdf [Accessed 3 Nov. 2017].

[15] Jandura, Louise. "Mars Science Laboratory Sample Acquisition, Sample Processing and Handling: Subsystem Design and Test Challenges" (PDF). *JPL*. NASA. [Accessed 3 Nov. 2017].

[16] *"MeArm Open Source Robot Arm (source files)"*. [Accessed 1 Nov. 2017].

[17] http://crustcrawler.com. (2017). *AX Robot Arms has high speed communication protocol with maximum kinetic accuracy constructed with aluminum*. [online] Available at: http://www.crustcrawler.com/products/AX%20Robotic%20Arm/ [Accessed 7 Nov. 2017].