



جامعة الأمير محمد بن فهد  
PRINCE MOHAMMAD BIN FAHD UNIVERSITY

College of Engineering

Department of Electrical Engineering

Fall 2018-19

Senior Design Project Report

**Nomad Rover**

In partial fulfillment of the requirements for the  
Degree of Bachelor of Science in Electrical Engineering

Team Members

	Student Name	Student ID
1	<b>ABDULLAH A. KAREEM</b>	<b>201401134</b>
2	<b>FAIEZ A. FEROZE</b>	<b>201400812</b>
3	<b>SAEED ALMUTAIRI</b>	<b>201302647</b>
4	<b>MOATH M. ALSOQAIR</b>	<b>201303884</b>

Project Advisors:

Advisor Name: **Mr. Ahmed Abul Hussain**

## *Abstract*

In recent years, humans have been able to incorporate robotics to perform various day-to-day tasks that were only possible by humans before. These robots can be seen in industries, medical fields, space programs and various other sectors. They were able to replace humans, providing better efficiency, as being able to carry out similar tasks without any fatigue. Besides the lack of knowledge in digital language, these robots are also restricted with poor utilization of hardware. Many of them were lacking a proper design, making them fragile, and unreliable. We are aiming at achieving a sturdy reliable design, that is not fragile and can hold up for a longer time, by making use of good materials like metal instead of plastics. Within the projects circuitry, we also are investing time for careful planning of components, sensors and hardware, so that they can be used efficiently.

With a reliable robust design, and easy to use interface, the Nomad Rover will be setting stones in making robots more efficient and easy to use in the military sector. With the ability to detect the surroundings and atmosphere for harmful substances, the Nomad Rover will be able to prevent soldiers to be caught in dangerous situations that may threaten their life.

In addition, with one to one control with hand gestures, Nomad Rover can provide unparalleled precision and accuracy for the controller to perform a multitude of operations, remotely in a safe place. Say in bomb diffusion operation, the bomb diffuser can maintain a safe distance from the bomb while diffusing, but can maintain the accuracy and precision, like doing with his own hands, thanks to the gesture controlled robotic hands. Besides the military sector, this level of control for arms can find application in other sectors, like in space programs, to explore much more efficiently, when sending unmanned drones to space. We believe Nomad Rover will be capable of carrying out various operations in dangerous locations and environments remotely, without risking the lives of hundreds of soldiers. We also look forward to feedback we receive from our instructors and colleagues, to improve the project in the future.

## Table of Contents

<i>Abstract</i> .....	2
1. Introduction.....	5
1.1 Project Definition.....	5
1.2 Project Objectives.....	5
1.3 Project Specifications.....	5
1.4 Product Architecture and Components.....	6
1.5 Applications.....	9
2. Literature Review.....	10
2.1 Project background.....	10
2.2 Previous Work.....	10
2.3 Comparative Study.....	11
3. System Design.....	12
3.1 Design Constraints.....	12
3.1.1 Design Constraints: Engineering Standards.....	12
3.1.2 Design Constraints: Environmental.....	12
3.1.3 Design Constraints: Safety.....	12
3.2 Design Methodology.....	13
3.3 Product Subsystems and Components.....	13
3.3.1 Product Subsystem1: Robotic Arm Controller.....	13
3.3.2 Product Subsystem2: Robotic Arm.....	14
3.4 Implementation.....	15
4. System Testing and Analysis.....	24
4.1 Subsystem 1: Communication for Control.....	24
4.2 Subsystem 2: Communication for camera.....	24
4.3 Subsystem 3: Robotic Arm Control.....	25
4.5 Overall Results, Analysis and Discussion.....	27
5. Project Management.....	28
5.1 Project Plan.....	28
5.2 Contribution of Team Members.....	29
5.3 Project Execution Monitoring.....	29
5.4 Challenges and Decision Making.....	30
5.5 Project Bill of Materials and Budget.....	31
6. Project Analysis.....	32
6.1 Life-long Learning.....	32
6.2 Impact of Engineering Solutions.....	32
6.3 Contemporary Issues Addressed.....	32
7. Conclusions and Future Recommendations.....	33
7.1 Conclusions.....	33
7.2 Future Recommendations.....	33
8. References.....	34

Appendix A: Progress Reports .....	35
Appendix B: Bill of Materials .....	44
Appendix C: Datasheets .....	45
Appendix D: Program Codes.....	50
Appendix E: Operation Manual.....	55

# **1. Introduction**

## **1.1 Project Definition**

We designed a robotic vehicle that can carry out tasks in environments that are inaccessible or unsafe. The robotic vehicle will be equipped with a robotic arm to perform various operations at a higher level of precision and accuracy.

## **1.2 Project Objectives**

- Robotic vehicle can handle various terrains and small obstacles
- Remote control with video feed.
- A robotic arm that follows the accuracy and precision of human hand.

## **1.3 Project Specifications**

The project will involve the design and testing of below mentioned features,

1. Two Degree of Freedom Mechanical arm with five fingers that will be remotely controlled using hand gestures
2. Robust vehicle chassis able to handle everyday abuse
3. Equipped with camera for video navigation
4. HC-12 wireless serial port communication module throughout the systems

## 1.4 Product Architecture and Components

- The entire project which was covered over two semesters, can be represented in the functional diagram in FIG 1.1,

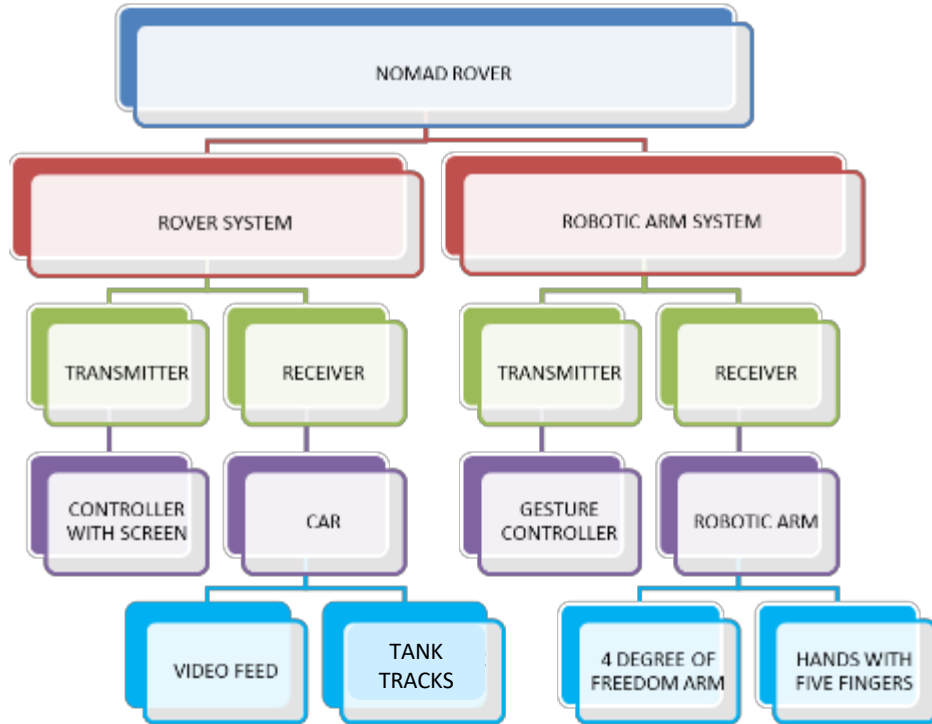


FIG 1.1 Functional Diagram of the Project

- The block diagram for the rover base and the controller is shown below in FIG 1.2,

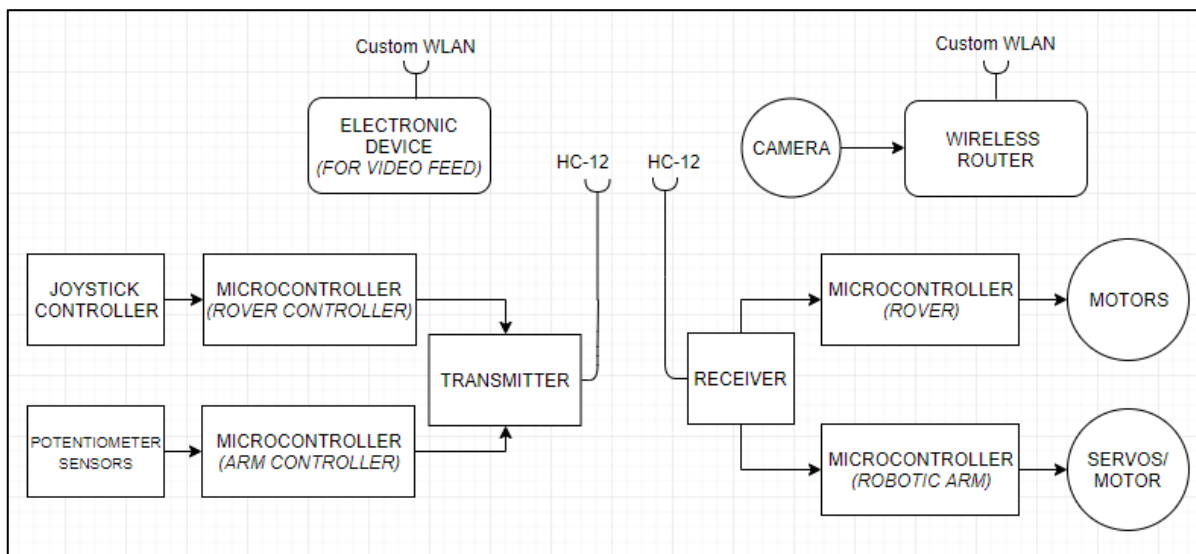
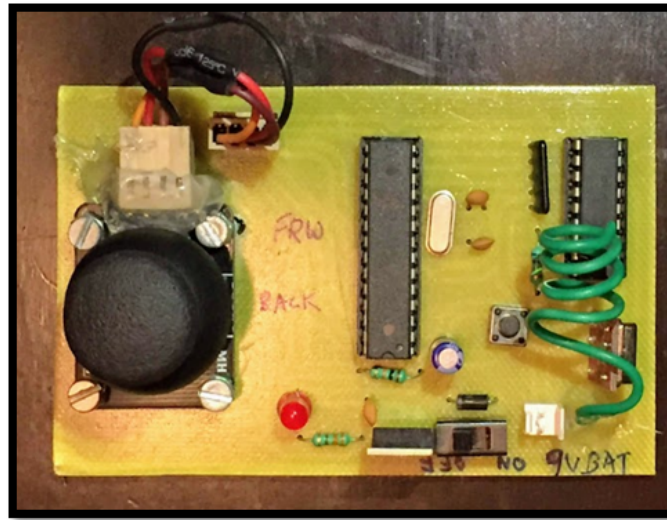


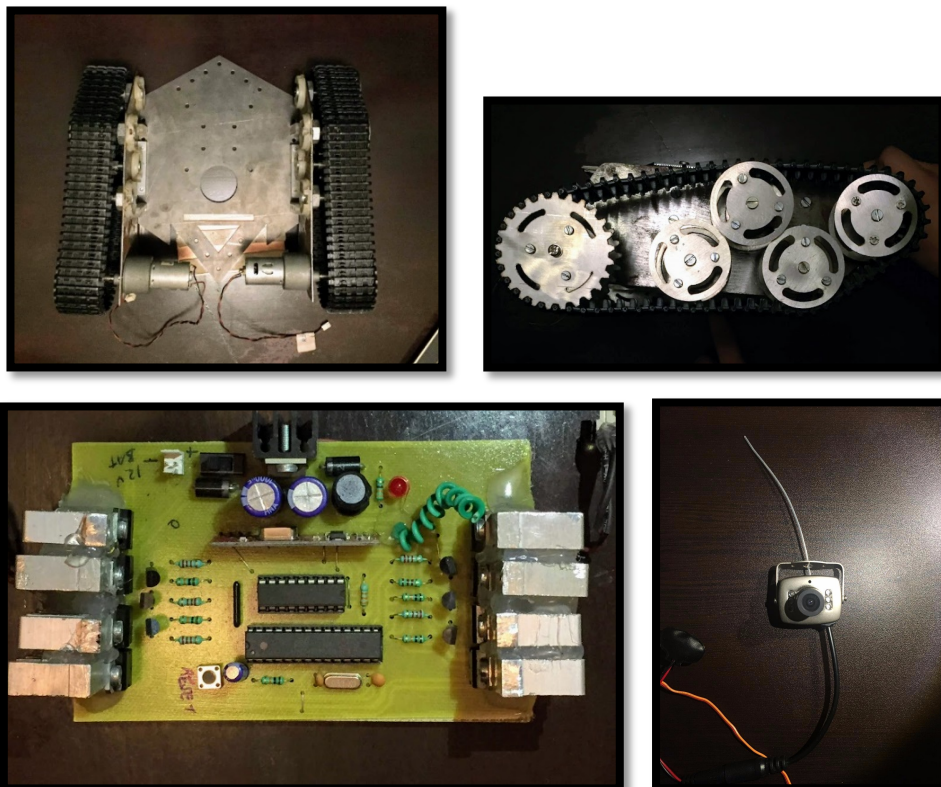
FIG 1.2 Block Diagram for Project

- Photos of real system

Phase I Rover System

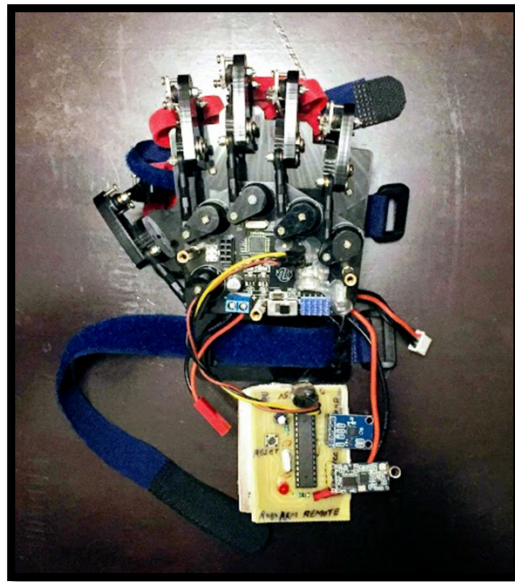


*FIG 1.3 Pictures for SS1-Rover Controller*

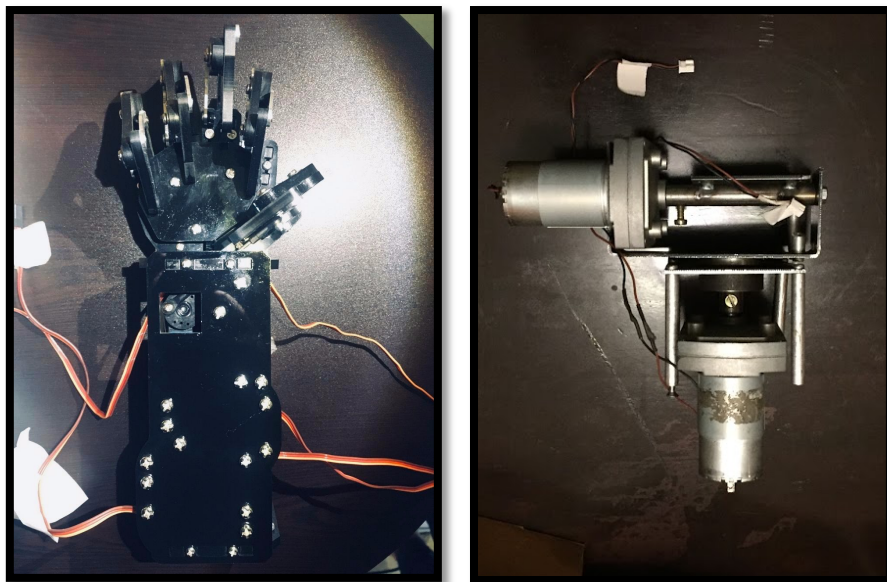


*FIG 1.4 Pictures for SS2-Rover Base*

Phase II Robotic Arm

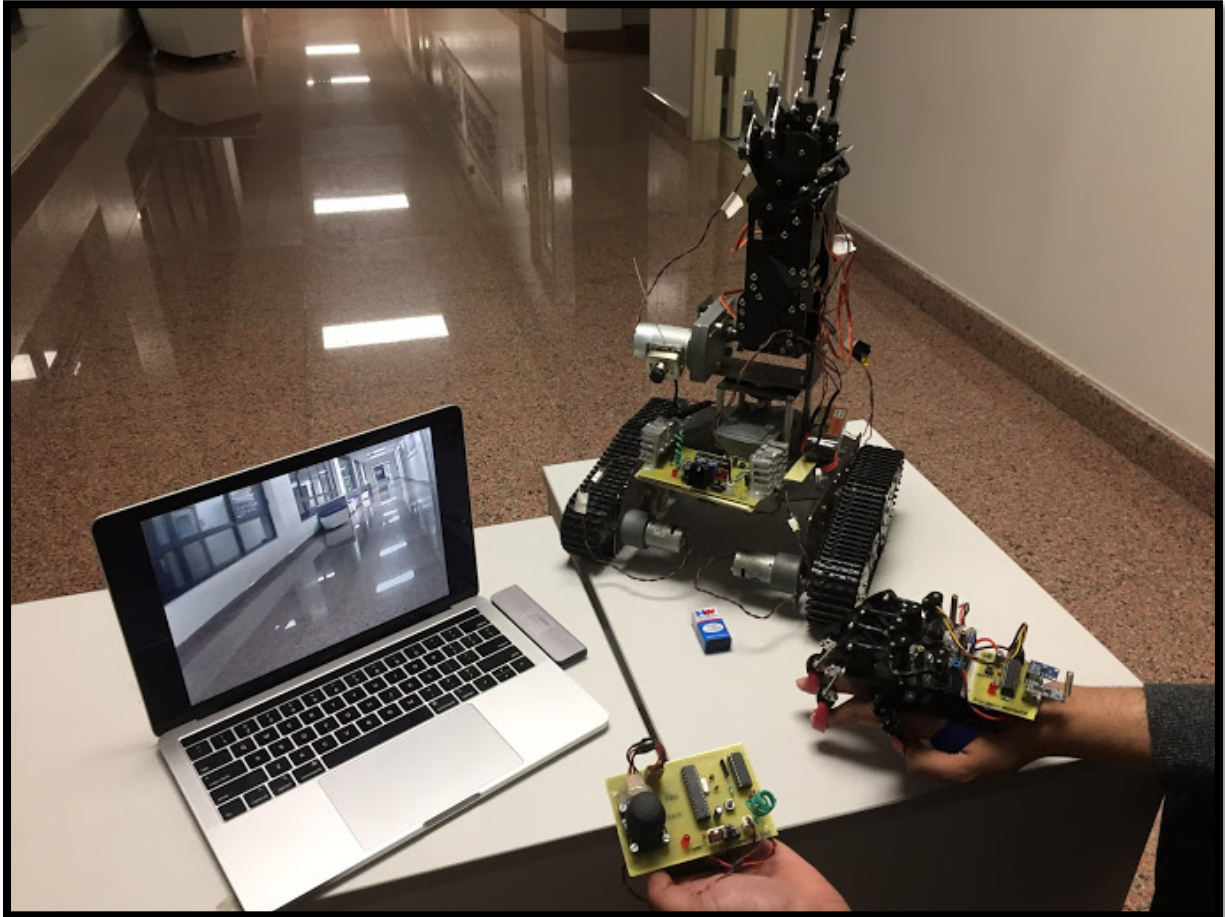


*FIG 1.5 Picture for SS1-Robotic Arm Controller*



*FIG 1.6 Pictures for SS2-Robotic Arm*

- For this semester we integrated both of the phases. We were able to complete Phase I during the Spring semester, and we worked on Phase II during the Summer semester. In the Fall semester we carried out remaining work in assembly, concentrated on testing and overall improvements and adjustments. The final assembly of both phases and both the systems is shown in FIG 1.7



*FIG 1.7 Nomad Rover*

## **1.5 Applications**

- The Nomad Rover can safely explore any unknown hostile region safely and anonymously remotely during hostile situations
- Nomad Rover can collect data or samples from inaccessible location
- The arm can assist astronauts or the army carry out small operations like transporting medical supplies or any other equipment in the battle field

## 2. Literature Review

### 2.1 Project background

- In the field of space exploration as of 2016, 18 astronaut and cosmonaut fatalities have been recorded related to space exploration. This is primarily because astronauts are at high risks in outer space due to limited resources. As such astronaut's space explorations are limited to planets that are closer to earth
- In the military sector combats with poisonous gases and likewise are beyond human senses to identify and has cost lot of lives. Also, risky situations like bomb disposal operations have a high failure rate that had cost lives.
- All of the above issues sheds light on the need for a remotely controlled drone that can perform simple tasks. Some of the previous projects and attempts, were able address to some of the issue but also faced a lot of drawbacks and limitations. One of the reasons was most of them didn't combine the rover and hand systems with each other. Our project on the other hand addresses the issue by combining two phases of a robotic hand and rugged base to solve most of the issue presented above

### 2.2 Previous Work

1. Oregon State Mars Rover, Jesse A. Grimes, Jesse A. Grimes, July 2011<sup>[1]</sup>

#### Features

- 6-wheel balloon tires with a motor for each wheel
- High ground clearance for off road driving
- Wireless video transmission
- Robotic arm structure with camera assistance

This project resembles closest to our project but implemented with different components

2. A Prototype Of Mars Exploration And Research Rover, Vishnu D. S, SCT College of Engineering, MARCH 2014<sup>[2]</sup>

#### Features

- Three wheels with two motors
- Atmospheric sensor
- Proximity sensor
- Solar powered
- Automated control

Although this project lacks a robotic hand, it is equipped with sensors, and is also self-powered with the help of the solar panel.

3. SENSOR CONTROLLED ROBOTIC HAND, Law Chor Hon, The University Of Hong Kong, January 2016<sup>[3]</sup>

Features

- 3D printed five finger hand/arm system
- Controlled using hand gestures
- Provides haptic feedback when grabbing objects

This previous project was selected as it shows the design and complexity involved in designing a gesture controlled robotic hand.

4. Gripper Design For A Modular Robot, Anna Maria Gil Fuster, Technical University Of Denmark, June 2015<sup>[4]</sup>

Features

- Claw style grip capable of grasping small objects
- 3D Printed claws to hold different shapes
- IR sensor to detect object and determine the torque

This project focuses on precision-controlled robotic hand that can be adjusted to hold objects of different shapes.

## 2.3 Comparative Study

In this section we have compared the features of the previous projects with ours as shown in the table below,

*Table 2.1 Comparative Study for Previous Projects*

<i>Features</i>	<i>PR 1</i>	<i>PR 2</i>	<i>PR 3</i>	<i>PR 4</i>	<i>EXP</i>
5 Finger Robotic Hand	✗	✓	✗	✗	✓
2 DOF Arm	✗	✓	✓	✗	✓
Precision Control Of Arm	✓	✓	✗	✗	✓
Live Video Feed	✓	✗	✓	✗	✓
Car Base	✗	✗	✓	✓	✓
Capable Of Covering Different Terrains	✗	✗	✓	✓	✓
Metallic Components For Car	✗	✗	✓	✓	✓
Sensors For Surroundings	✗	✗	✓	✓	✓
Remote Control	✗	✗	✓	✗	✓

## 3. System Design

### 3.1 Design Constraints

#### 3.1.1 *Design Constraints: Engineering Standards*

- All of the components used in this project communicates through the HC-12 wireless serial port. The HC-12 module uses a Silicon Labs Si4463 to provide the RF communications link. This is a high performance, low current, single-chip “EZRadioPRO” family, and standardized under FCC<sup>[5]</sup> (I-PEX 20279-001 -E-01)
- Arduino ICs used in the project for either subsystems are licensed under the GNU General Public License (GPL)<sup>[6]</sup>, which allows user freely run and edit the software.
- Also, the ATmega32P processor within the circuitry, which belongs to Microchip is certified under ISO/TS-16949<sup>[7]</sup>

#### 3.1.2 *Design Constraints: Environmental*

- Stainless steel chassis used in subsystem 1 is recyclable compared to chassis made primarily of plastic
- Most of the testing in software were carried out in USB power, in order to minimize the use of batteries
- Also, in the event of requirement mobile DC power, rechargeable Li-Po batteries were used, to minimize the disposal of batteries
- Voltages and currents were carefully monitored, in order to prevent wastage

#### 3.1.3 *Design Constraints: Safety*

- Insulation was provided wherever electric components made contact with stainless chassis
- The voltages of components in the subsystem were carefully noted in order to prevent over supply which can damage the components
- Capacitors are Inductors were used as protection for the sensors and other components in the circuits from shorts, over currents/voltages.

## 3.2 Design Methodology

### 3.3 Product Subsystems and Components

#### 3.3.1 Product Subsystem1: Robotic Arm Controller

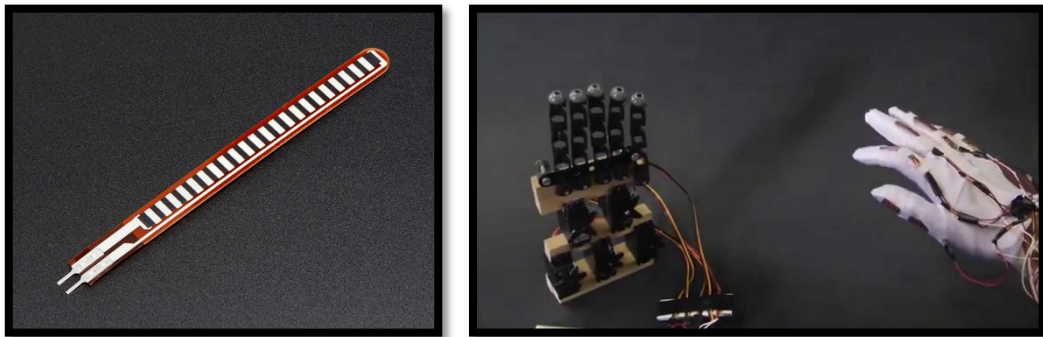
##### Sensors

The main aim for the sensors where to track the movement for each finger, and to convert it to a digital signal in order to be processed and transmitted to the robotic hand. We considered the following alternatives when selecting a sensor.

##### - Flex Sensors

Initially we worked with flex sensors, as it was a popular choice in this scenario. Even though it functioned we had a few drawbacks,

- Constant calibration for different hands
- Complex coding involving multiple variables
- High sensitivity



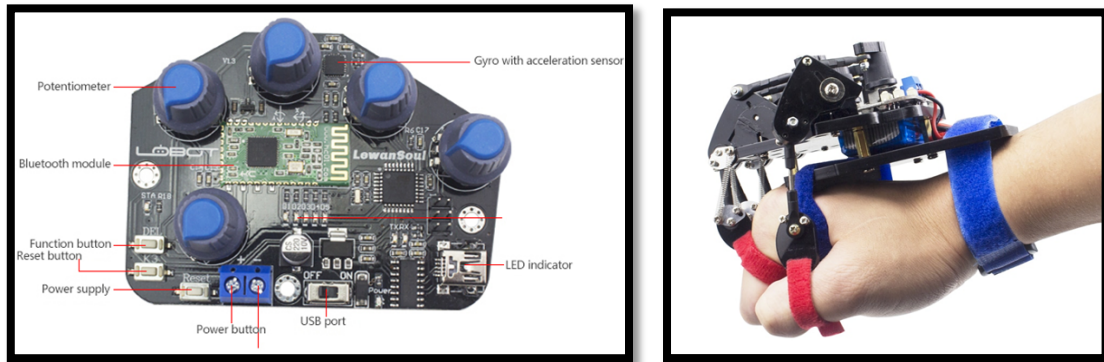
*FIG 3.1 Implementing Flex Sensor for SSI*

Due to these drawbacks associated with the flex sensors we explored for another alternative in order to address these problems

##### - Potentiometer

We have worked with potentiometer consistently in our previous courses, and we were familiar with the coding.

- With a design we found, each potentiometer can be attached to a finger, and the motion of the finger can be tracked regardless of shape and size differences
- The motion of the finger is converted into analog readings by the potentiometer, making it much easier to code with only single variable to work with.
- With adequate sensitivity and precision, we think the potentiometer is a more efficient sensor for our controller.



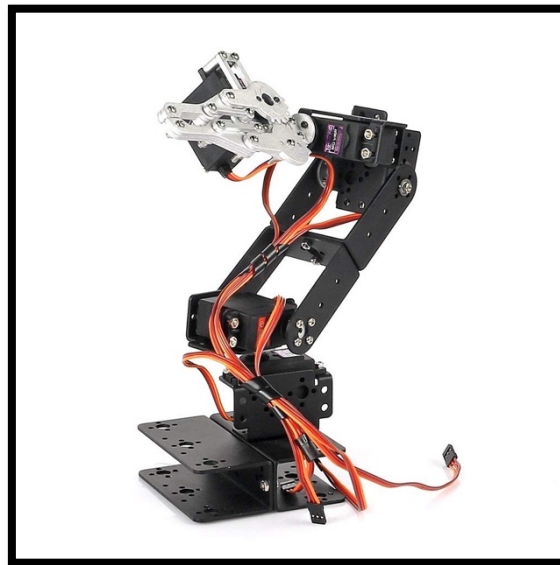
*FIG 3.2 Implementing Potentiometer for SS1*

### **3.3.2 Product Subsystem2: Robotic Arm**

#### Robotic Arm Hardware

The main aim for the robotic arm hardware, was to be able to perform functions similar to a human hand. The required specifications were claws/fingers design with 2 degree of freedom motion support

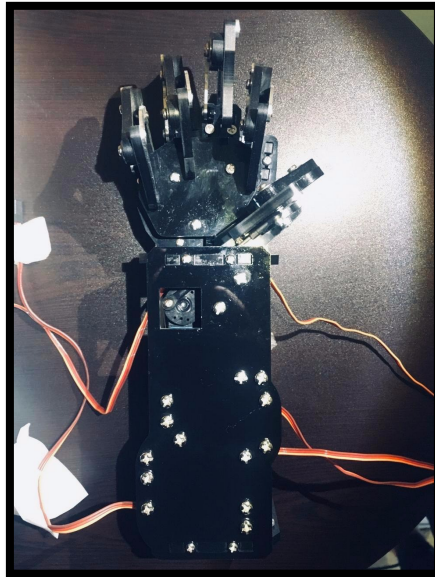
- SainSmart 5-Axis Robotic Arm



*FIG 3.3 SainSmart 5-Axis Robotic Arm for SS2*

In the first semester we worked with the SainSmart 5-Axis Robotic Arm as shown in the picture above. With four degrees of freedom, and a clamp style grip, this hardware was close to our specifications. We carried out various experiments with coding and control. But as per our specification we also needed a 5-finger robotic hand design, in order to imitate a human hand.

- Industrial Robot Arm Bionic Robot Hands



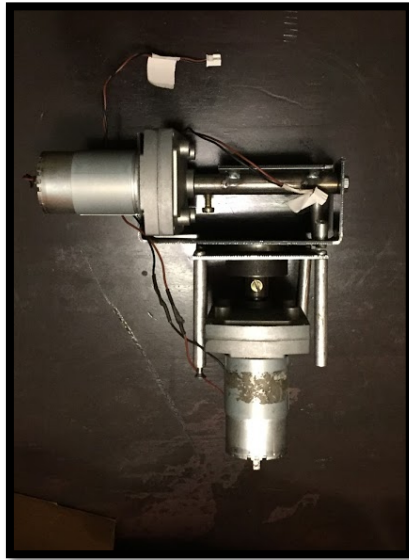
*FIG 3.4 -Industrial Robot Arm Bionic Robot Hands and custom base for SS2*

### **3.4 Implementation**

#### System Integration

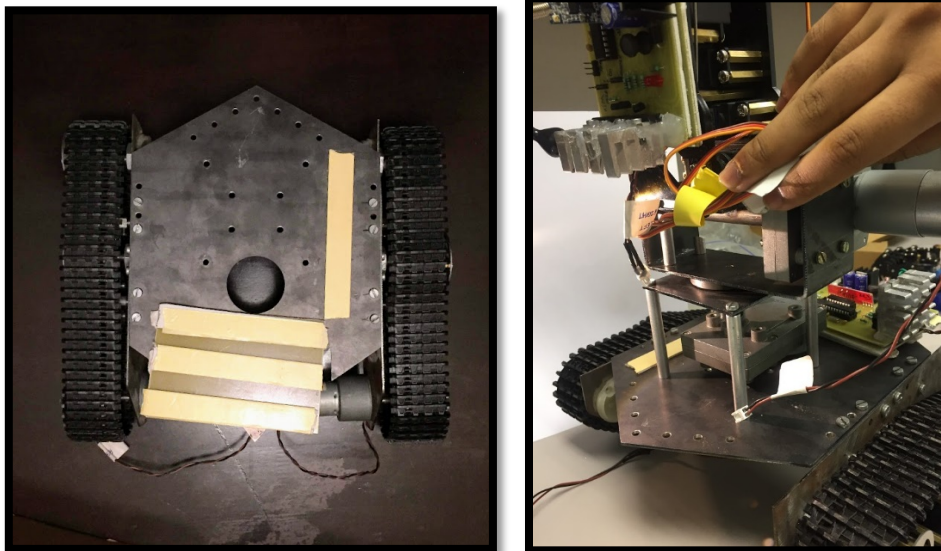
In order to attach the robotic arm to the rover base, a custom component called *the robotic arm base* was designed to attach the robotic arm to the rover base. It also helps the robotic arm to lean forward and backward. The *robotic arm base* consists of two metallic clamps and two DC motors, attached to gears.

The gears help in functioning the DC motors as servos, while preserving the torque. This way we were able to efficiently move the arm forward and backward, or rotate left and right, without any stutter or lag. The robotic arm base itself had a DC motor at the bottom to insert into the rover base in order to perfectly secure robotic arm to the rover base



*FIG 3.5 Custom designed Robotic Arm Base*

The position of robotic arm on the rover base was another challenge when integrating the two systems. The robotic arm should be able to move around freely forward or backward without the base tipping over. In order to do so the robotic arm base was placed towards rear of the chassis. This way the base would be steady and also allow robotic arm to reach farther from the car.



*FIG 3.6 Robotic Arm Position on Rover Base*

## Communication

Selecting the right communication protocol has been a challenge from the beginning of the project. We were looking for the following when selecting a communication channel,

- Reliability, and easy setup
- Adequate range
- Easy coding

We came across a lot of communication media, some of them were taught to us in various courses, and others we explored, as shown below,

- Bluetooth  
Bluetooth was the first communication channel we worked with. With support to any device with Bluetooth, and easy pairing, also had a few drawbacks,
  - Short Range (<10M)
  - Can only be configured as a receiver, not ideal for controllers
- XBee  
We also explored XBee protocol for our project. With a long range of nearly a kilometer and configurable modes as a transmitter or a receiver we faced the following problems
  - Complex coding and programming through X-CTU software
  - Connection requires XBee shields, which were hard to acquire.
- HC-12  
HC-12 wireless serial port communication module is a new-generation multichannel embedded wireless data transmission module used in a lot of modern day electronics like POS system, Wireless acquisition of gas meter data, Vehicle keyless entry system.
  - Long Range (1000M in an open space)
  - Three working modes, adapting to different application situations
  - Built-in MCU performs communication with external device through serial port, no programming or configuration required for basic use
  - Easy connections

So we implemented HC-12 throughout different components in our project working in different channels.

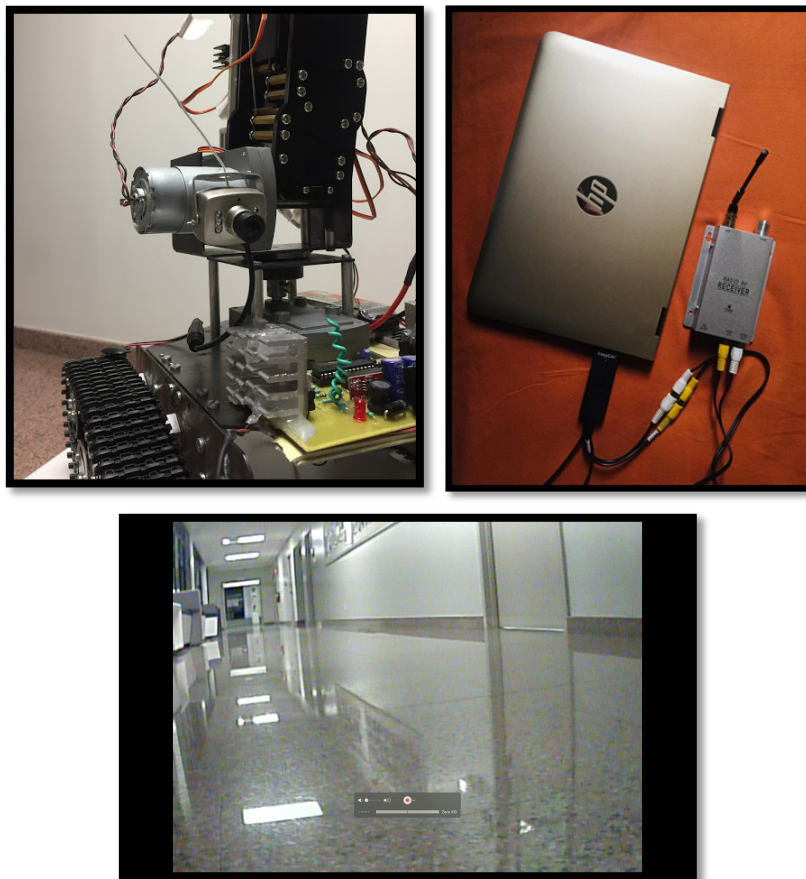
## Camera

We started the project with a raspberry camera. With a raspberry processor connected to a any network, it can be programmed to transmit any video feed to an electronic device like a smartphone or a computer, when connected to camera. This gives a long range, being connected to a network. But the camera hardware is the reason we had to look for alternatives. It transmitted low quality videos at 480p, with very long delays of around five seconds, without night vision. The screenshots of the footage and camera is shown below,



*FIG 3.8 RaspiCam and footage*

After a thorough research we found the generic infrared camera. With in built processor that is pre-programmed, and a higher quality footage with much lower lag, and night vision it clearly had an advantage over the RaspiCam. The only drawback was in the range, which was around 20-30m. but even this range is practical, and can be improved by increasing the voltage of the antenna. The setup also consists of router that receives the video feed to a PC,



*FIG 3.9 Infrared Camera setup and footage*

## Circuitry

In our project we decided to go with PCB (printed circuit board). The main advantage of implementing this style of circuitry, is the removal of any physical jumper wires. All the connections were printed digitally, and this ensures stable connections.

Before printing the PCB, a PCB layout was made in a free software called **pcbWEB**. The designed PCB layouts are shown below along with the circuit diagrams,

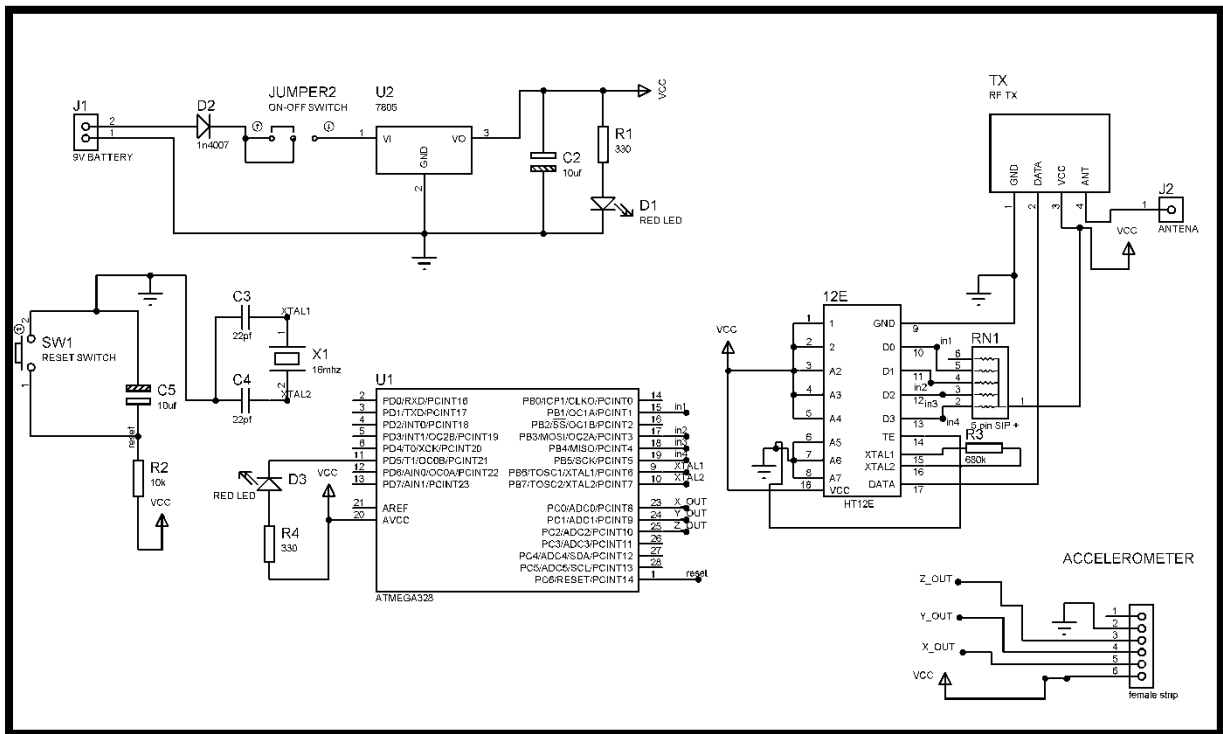


FIG 3.7 Circuit Diagram of Robotic Arm Controller

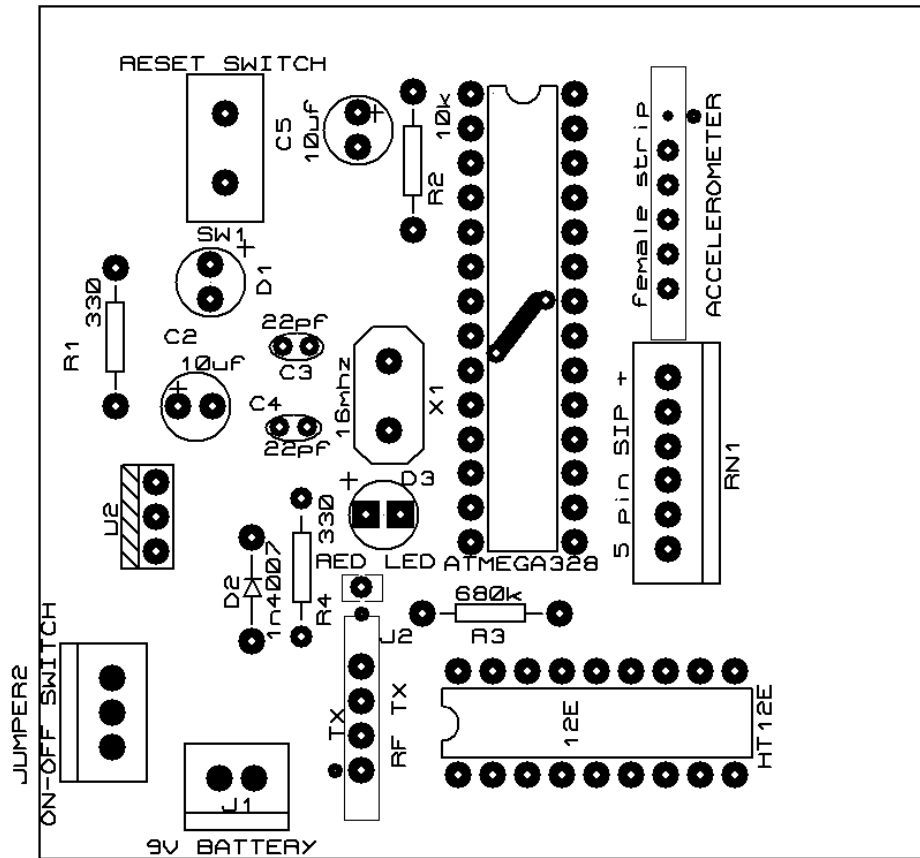


FIG 3.8 PCB layout of Robotic Arm Controller

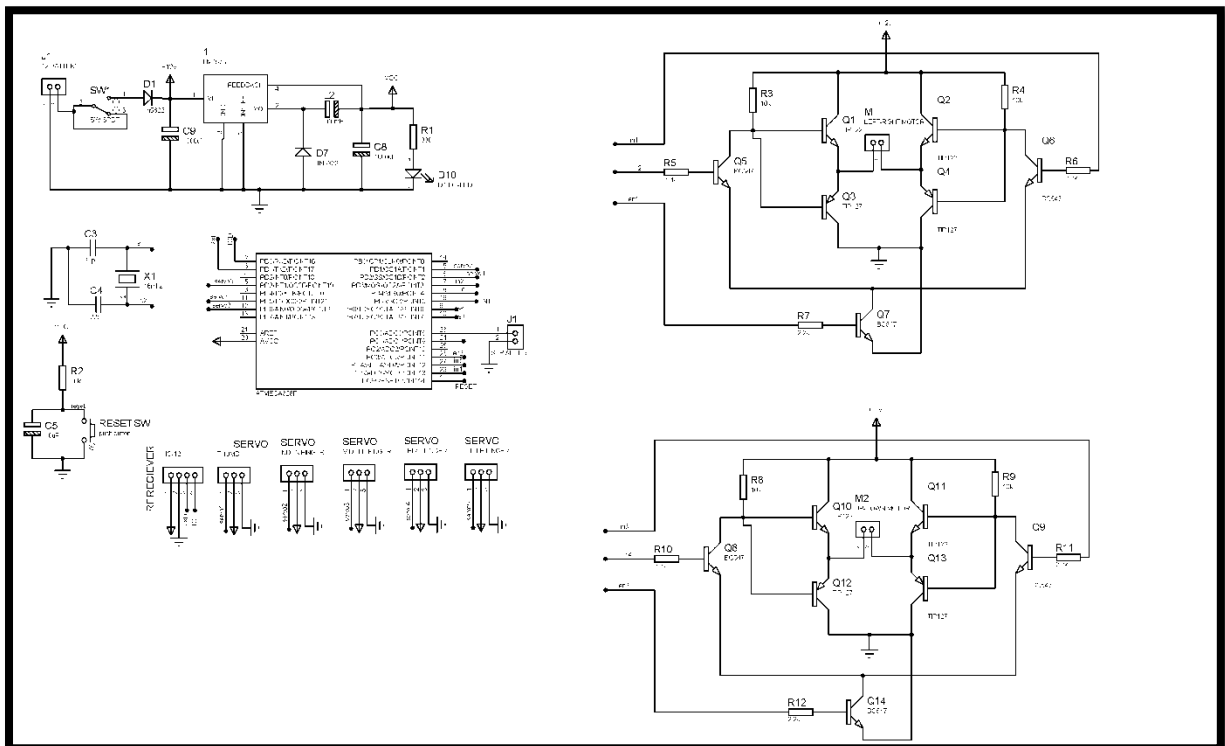


FIG 3.9 Circuit Diagram of Robotic Arm

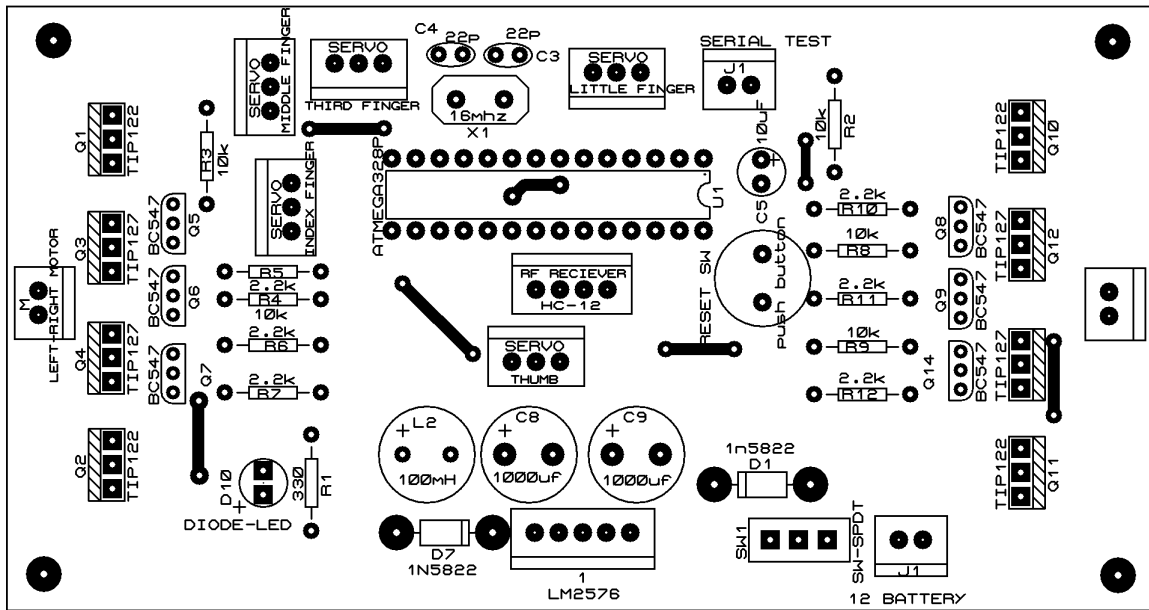


FIG 3.10 PCB layout of Robotic Arm

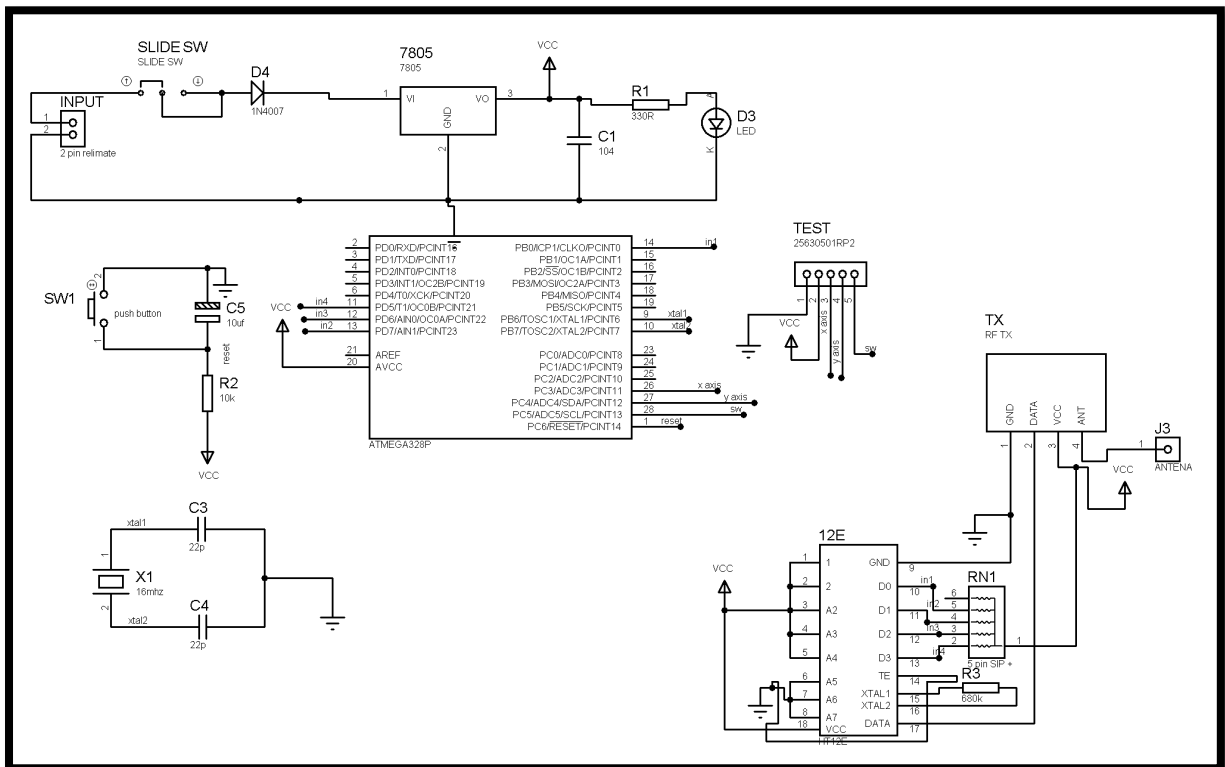


FIG 3.11 Circuit Diagram of Rover Controller

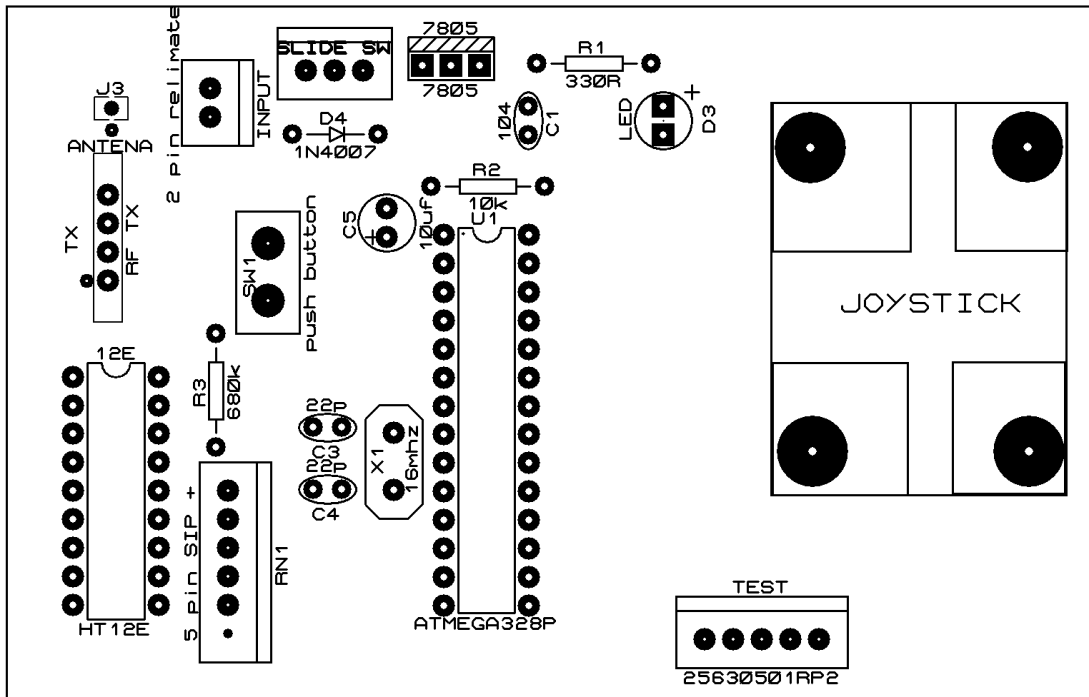


FIG 3.12 PCB layout of Rover Controller

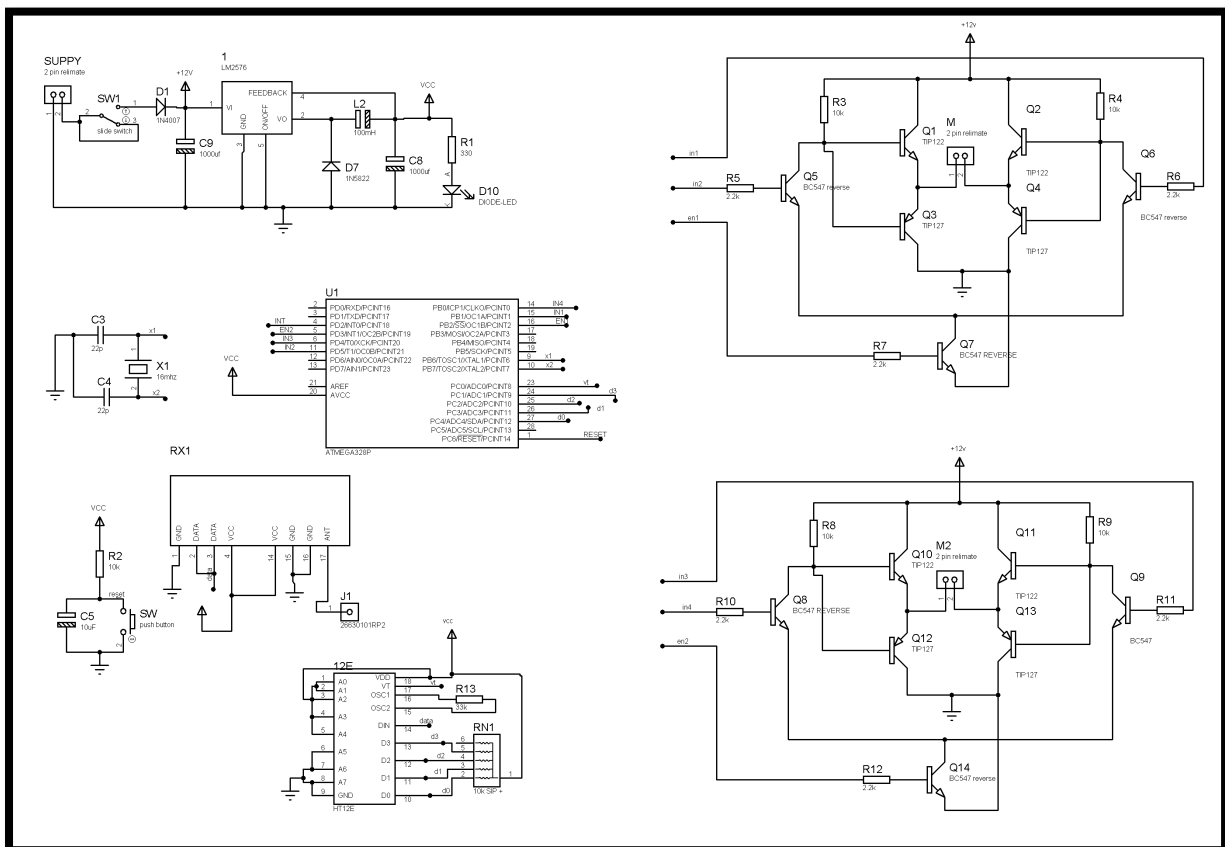


FIG 3.13 Circuit Diagram of Rover Base

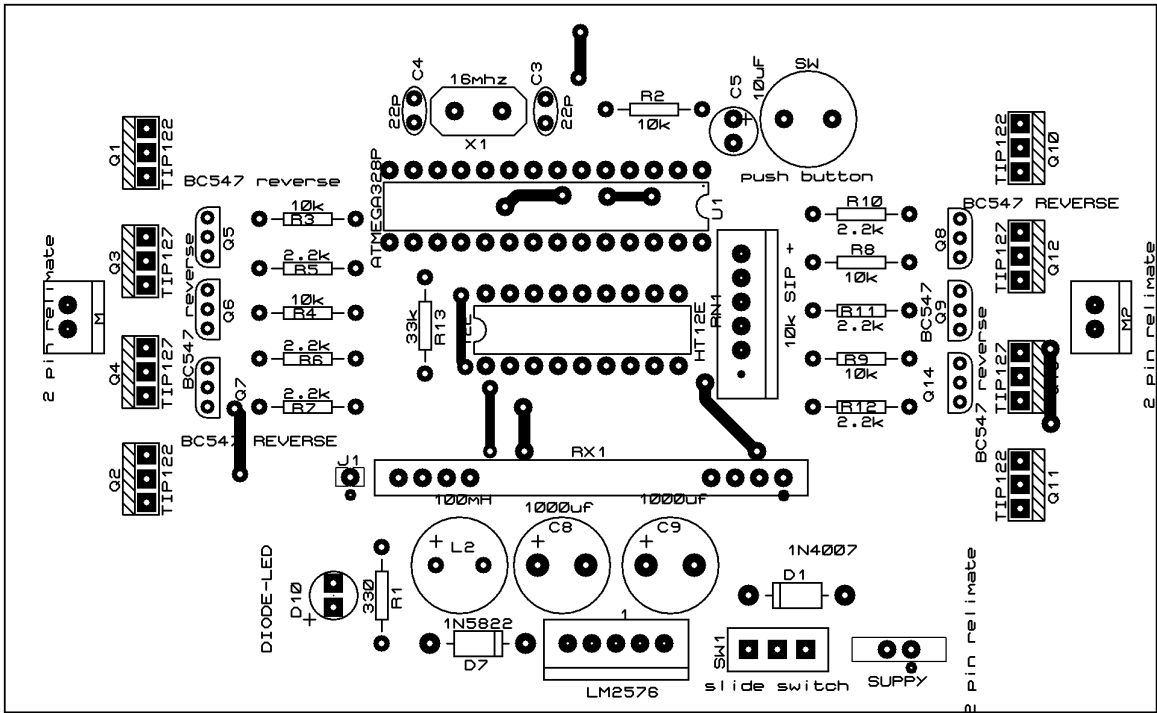


FIG 3.14 PCB layout of Rover Base

## 4. System Testing and Analysis

### 4.1 Subsystem 1: Communication for Control

#### Objectives:

The objectives for this testing the communication for control are:

1. Communication aspects between robotic hand and controller
2. Communication aspects between rover base and controller

#### Setup:

- For all our range testing in the project we used *Stanley TLM-165* as shown in the figure below.



FIG 4.1 Stanley TLM-165 during testing

- In order to carry out our testing we choose various scenarios of indoor environments and outdoor open spaces, in the university

#### Results:

Table 4.1 Testing Results Communication for Control

<i>Component</i>	<i>Results</i>	<i>Location</i>
<i>Robotic Hand</i>	Solid Connection indoors <b>(tested 300m)</b>	Indoors
	Solid Connection outdoors <b>(tested 500m)</b>	Outdoors
<i>Rover</i>	Solid Connection indoors <b>(tested 25m)</b>	Indoors
	Solid Connection outdoors <b>(tested 25m)</b>	Outdoors

### 4.2 Subsystem 2: Communication for camera

## Objectives:

The testing objectives for camera communication includes,

1. Communication Range between camera and receiver
2. Quality of video feed and audio
3. Night vision compatibility

## Setup:

In order to carry out the testing the following equipment were used

- Stanley TLM-165 Laser Distance Measurer
- HP-Pavilion Laptop connected to Radio AV Receiver through Easycap converter, with honestech VHS to DVD 3.0 software, as shown in the figure.



FIG 4.2 Laptop Setup during testing

## Results:

Table 4.2 Testing Results Communication for camera

<i>Component</i>	<i>Parameter</i>	<i>Results</i>
<i>Camera Communication</i>	Range	Indoors: 20 m Outdoors: 30 m
	Night vision	Functioning infrared vision in the dark
	Pairing Process	Instant pairing when turned on and in range

## 4.3 Subsystem 3: Robotic Arm Control

### Objectives

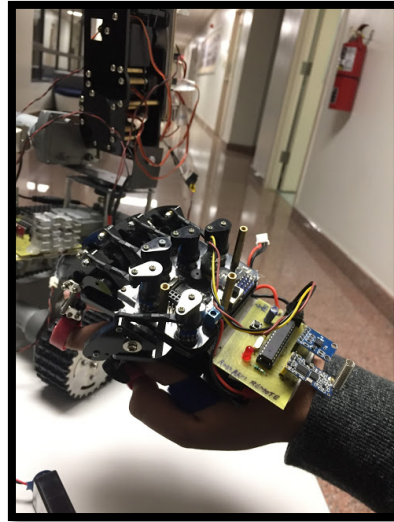
The testing objectives for robotic arm,

1. Precision/Accuracy of hand movements

2. Load capability
3. Precision/Accuracy of arm movements

### Setup

The testing for arm movements and finger accuracy were tested by visual inspection of the project as shown below



*FIG 4.3 Visual Inspection of Robotic Hand and Arm*

### Results

Table 4.3 Testing Results Robotic Arm Control

<i>Component</i>	<i>Parameter</i>	<i>Results</i>
<i>Parameter</i>	Precision	Finger to Finger Accuracy
	Arm Movement	Accurate Position tracking
	Delays	0.5 Sec Arm Movements 0.2 Sec Finger Tracking

## 4.5 Overall Results, Analysis and Discussion

Shown below is an overall analysis of various parameters in both the subsystems,

*Table 4.5 Testing Results*

	<i>PARAMETER</i>	<i>RESULTS</i>	<i>COMMENTS</i>
<b>ROVER BASE</b>	Range (Communication)	25 m tested (indoor/campus) 25 m (outdoor/open space)	Tested in campus from robotics lab to outside Mosque
	Movements (Forward/Reverse/L/R) (Circuitry)	Speed = 0.2 m/s Forward, right and left are accurate, reversing has a slight deviation due to design aspect	Tested in campus on marble terrain
	Battery Life	At controller: 30 min. runtime At receiver: 15 min. runtime	No comments
	Camera	Range of operation recorded, 20 m Indoors 30 m Outdoors Instant pairing when turned on	Tested in campus and also in dark room for night vision
<b>ROBOTIC HAND</b>	Precision/Accuracy/Movements (Circuitry)	Precise movements for each finger Lag set for 0.2s	Visual inspection
	Range (Communication)	300 m tested (indoor/ campus) 500 m (outdoor/open space)	Tested in campus from robotics lab to outside Mosque
	Battery Life	At controller: 30 min. runtime At receiver: 30 min. runtime	No comments
	Load Holding/ Reachability (Design)	Reachability: can reach objects of 25cm from the ground Each finger can comfortably 150 gm object, and hand can lift an object weighing the same	Each finger tested to hold and lift an iPhone 6 plus (~150gm)
	Pairing Process (Communication)	Instant pairing	When switched on and in range all the components are connected

## 5. Project Management

### 5.1 Project Plan

Table 5.1 project management plan

<i>SN</i>	<i>Tasks &amp; Responsibilities</i>		<i>Begin</i>	<i>End</i>	<i>% Completion</i>
1	Finish design subsystem SS1 ( <b>ROBOTIC ARM</b> )	All	Sep 10	Sep 17	100%
2	Implement Design for SS1 (Hardware Assembly and wiring)	AK, SM	Sep 10	Oct 01	100%
3	Test and refine design for SS1 (Holding objects and Motion test)	FF, SM	Oct 01	Oct 15	100%
4	Design subsystem SS2 ( <b>ROBOTIC ARM CONTROLLER</b> )	All	Sep 10	Sep 17	100%
5	Implement Design for SS2 (Communication channel calibration, wiring)	AK, FF	Sep 10	Oct 01	100%
6	Test and refine design for SS2 (Range Testing, gesture recognition)	FF, MS	Oct 01	Nov 15	100%
7	Integrate subsystems SS1& SS2	All	Oct 15	Nov 20	100%
8	Test and make final changes	All	Oct 22	Nov 29	100%
9	Write Progress Reports (Weekly)	All	Sep 24	Nov 22	100%
10	Prepare Midterm Presentations	All	Oct 01	Nov 12	100%
11	Prepare Demo. (Prototype)	All	Sep 10	Dec 10	100%
12	Prepare Final Presentation	All	Oct 22	Dec 10	100%
13	Prepare Final Report	All	Sep 10	Nov 19	100%

## 5.2 Contribution of Team Members

Table 5.2 Contribution of Team Members

<i>Task</i>	<i>AK</i>	<i>FF</i>	<i>SM</i>	<i>MS</i>	<i>Task Total</i>
Research And Planning	25%	25%	30%	20%	<b>100%</b>
Acquiring Components	25%	30%	20%	25%	<b>100%</b>
Assembly	20%	35%	20%	25%	<b>100%</b>
Coding	30%	20%	35%	15%	<b>100%</b>
Reports and Presentation	35%	20%	30%	15%	<b>100%</b>
Testing and Improving	25%	30%	25%	20%	<b>100%</b>

## 5.3 Project Execution Monitoring

Table 5.3 Meetings with Advisors

<i>S.no</i>	<i>Minutes</i>	<i>Date</i>
1.	Semester Plan: Tasks for the final semester, management plan, project progress and review	26.09.2018
2.	Technical Meeting: - Alternatives for communication from XBee - Raspberry Camera system replacement	24.10.2018
3.	Presentation Feedback: slides to add/remove, image re-edit, content check	12.11.2018
4.	Technical Meeting: Review of progress	19.11.2018
5.	Testing meeting: Plans for testing – components to test and testing procedures	26.11.2018
6.	Final presentation meeting: - What to present - Changes to make in Midterm Presentation	03.12.2018

Table 5.4 Team Meetings

<i>S.no</i>	<i>Minutes</i>	<i>Date</i>
1.	Semester Plan: Tasks for the final semester, management plan, project progress and review	26.09.2018
2.	Assembly Meeting: - Alternatives for communication from XBee - Raspberry Camera system replacement	24.10.2018
3.	Presentation Meeting: slides to add/remove, image re-edit, content check	12.11.2018
4.	Assembly Meeting: Assemble both subsystem	21.11.2018
5.	Testing Meeting 1:	06.12.2018
6.	Testing Meeting 2:	09.12.2018
7.	Video shoot meeting:	10.12.2018
8,	Final Presentation Meeting:	11.12.2018

## 5.4 Challenges and Decision Making

- Camera  
Initially, we used the Raspberry Pi 3 with a camera attachment kit as our primary live video streaming solution, eventually after a lot of testing we decided to move on from the raspberry Pi 3 and add a new infrared camera instead. We decided to switch to the infrared camera because it had a better video resolution and also a reduced delay on the live feed which is a very important factor. The average delay time on the raspberry pi 3 was five seconds, the delay on the IR camera was less than one second. Another reason was the ability to switch from day mode to night-vision mode using its built in Infrared sensors. These were the 2 main reasons why we decided to switch from the Raspberry Pi 3 to a generic infrared camera.
- Communication HC12 For Rover Replaced With RF  
The communication system on the Robotic Vehicle burned out a day before the demo, since we were unable to find the required HC-12 component to replace the burned out one, we have to use RF as our backup option. It was initially hard to set it up within the given time, but after a lot of research we were able to completely set up the RF system to control the robotic vehicle. Since HC-12 provides us with a longer range, the HC-12 is still our primary option and we plan on adding the HC-12 after we receive the new components again and to have the RF as a backup option in case of any other incidents.
- Component Availability  
As mentioned in the earlier paragraph (refer to 5.4 – Communication for HC-12 replaced by RF), It was extremely difficult to obtain new HC-12 components in the limited time

we had, due to budget constraints, we were unable to buy spare components in advance, this isn't just limited to the HC-12 but to all other components as well. We initially purchased the exact number components required for our project without buying any spares. Around 80% of all the equipment and components needed were purchased from outside the kingdom of Saudi Arabia, with an average shipping duration of two weeks for each package.

- Battery

Battery life was also a major issue as we had to use a smaller battery due to the limited space on the Tank Chassis. Due to this, we had to settle with a smaller 2100 mAh 11.1 V Li-Po Battery system. This provided us with an average battery life of upto thirty minutes. In the future we can further upgrade the chassis to increase space and use a bigger battery system in order to increase the overall battery life of the Nomad Rover.

## 5.5 Project Bill of Materials and Budget

For Phase I, the budget we set was 8000 SAR. But after receiving the products we were able to acquire projects through deals and discounts thereby helping us stay lower than the budget. The estimated costs of the materials have been given below. Please note all prices mentioned below are in *Saudi Arabian Riyals (SAR)*

*Table 5.2 Budget Estimate*

<b>S.No.</b>	<b>Description</b>	<b>Quantity</b>	<b>Unit Cost (SAR)</b>	<b>Total Cost (SAR)</b>
1	ATMEGA 328 + Programming Kit	6	200	1200
2	Stainless Steel Car Chassis (with motor)	1	1200	1200
3	Camera Kit	2	400	800
4	HC-12	5	35	175
5	4 DOF Arm and hand	1	1300	1300
6	Assorted LED kits	2	100	200
7	Controller (Joystick)	1	50	50
8	Battery System (Li-Ion- LiPo)	5	60	300
9	Miscellaneous	-	-	1000
<b>TOTAL</b>				<b>6225 SAR</b>

# 6. Project Analysis

## 6.1 Life-long Learning

This project was a major learning experience for us regarding electrical engineering. We were able to earn a lot of experiences, skills, knowledge and learnt how to overcome a lot of challenges that we will need in our future career. Some of the notable ones include,

- During the course of implementing Phase I, we came across a lot of new hardware that we have not worked with before. Some of them like Arduino microcontrollers, DC motors, a few sensors were covered in syllabus, whereas some other components like Raspberry, XBee, gas sensors, joystick controller had to be learnt externally with online articles and tutorials.
- Although IDE was the most used software, we still had to work with software including X-CTU, Raspbian OS, TINKERCAD in order to carry out programming and simulation
- Working on this project helped us in gaining a lot of project management skills. Some of them include recording data, experimenting, producing weekly reports, updating progress, and dividing work between the team.
- Time was limited we had to make use of it efficiently. This was where the project plan that was used since 3<sup>rd</sup> Week, helped us in achieving our tasks and completing Phase I in time

## 6.2 Impact of Engineering Solutions

Technology has always played a role in making a man's life easier and has also become an important aspect of human evolution. And as such engineering has been in forefront in solving real world problems in the society.

We believe with the goals and specifications we had set for the project and problems we are willing to address, our project can also be part of that revolution in safeguarding lives, carrying out explorations, assisting in healthcare and also as an efficient worker in industries.

## 6.3 Contemporary Issues Addressed

Technology has always played a role in making a man's life easier and has also become an important aspect of human evolution. And as such engineering has been in forefront in solving real world problems in the society.

We believe with the goals and specifications we had set for the project and problems we are willing to address, our project can also be part of that revolution in safeguarding lives, carrying out explorations, assisting in healthcare and also as an efficient worker in industries.

# 7. Conclusions and Future Recommendations

## 7.1 Conclusions

In conclusion we are proud to be able to complete the project in time. Several improvements and trials were carried out in order to be at this point. This was possible because each member completed his responsibility as per the plan, and also produced results.

Unlike the previous projects we had mentioned in section 2.2, our project is unique in many ways. It's primarily based on Arduino and also uses XBee for communication. This way the project is much simpler to program and also related to our course.

Moath:	Challenges:	Researching components, Team Work
	Experience:	Block diagram, teamwork, component assembly
Saeed:	Challenges:	Power management, Troubleshooting, sensors
	Experience:	XBee, HC12 Programming and testing
Faiez:	Challenges:	Designing subsystems, component assembly
	Experience:	Raspberry programming, Python Coding
Abdullah:	Challenges:	Coding Logic, Time management
	Experience:	Planning, XBee HC12 Programming and testing

## 7.2 Future Recommendations

We also had other ideas we had in mind to contribute to the Phase I of the project

- Experiment on communicating between Arduinos using Bluetooth modules.
- Use NI MYRIO for video streaming
- Increase the runtime of the project by installing higher capacity batteries.
- Metal constructed hand, capable of holding heavier objects.
- Improving range of camera, by powering antenna with higher voltage.

## 8. References

- [1] Robson, A. and Seward, E. (2018). SAE Technical Paper Series. [online] oregonstate.edu. Available at:  
<http://web.engr.oregonstate.edu/~shumanm/MarsRover/2011DesignReport.pdf> [Accessed 4 May 2018].
- [2] On, A. (2018). A PROTOTYPE OF MARS EXPLORATION AND RESEARCH ROVER. [online] Wikimedia.org. Available at:  
[https://upload.wikimedia.org/wikipedia/commons/d/de/Mars\\_Exploration\\_and\\_Research\\_Rover\\_%28\\_Prototype\\_%29\\_Report.pdf](https://upload.wikimedia.org/wikipedia/commons/d/de/Mars_Exploration_and_Research_Rover_%28_Prototype_%29_Report.pdf) [Accessed 4 May 2018].
- [3] Lundborg, G. (2018). The Hand and the Brain. [online] Available at:  
[http://i.cs.hku.hk/fyp/2016/report/detailed\\_intermediate\\_report/fyp16036-LAW%20Chor%20Hon\\_11256049\\_assignsubmission\\_file\\_intermediate-report.pdf](http://i.cs.hku.hk/fyp/2016/report/detailed_intermediate_report/fyp16036-LAW%20Chor%20Hon_11256049_assignsubmission_file_intermediate-report.pdf) [Accessed 4 May 2018].
- [4] Lafifi, M. (2018). Researchgate.net. [online] Available at:  
[https://www.researchgate.net/profile/Mohamed\\_Mourad\\_Lafifi/post/How\\_is\\_gripper\\_for\\_ce\\_control\\_currently\\_being\\_used\\_to\\_compensate\\_for\\_moments\\_due\\_to\\_gripping\\_at\\_a\\_distance\\_from\\_an\\_objects\\_centre\\_of\\_mass/attachment/59d63b0679197b8077998142/AS%3A408645595615235%401474440110515/download/Gripper+design+and+development+for+a+modular+robot+AnnaMariaGil-report.pdf](https://www.researchgate.net/profile/Mohamed_Mourad_Lafifi/post/How_is_gripper_for_ce_control_currently_being_used_to_compensate_for_moments_due_to_gripping_at_a_distance_from_an_objects_centre_of_mass/attachment/59d63b0679197b8077998142/AS%3A408645595615235%401474440110515/download/Gripper+design+and+development+for+a+modular+robot+AnnaMariaGil-report.pdf) [Accessed 4 May 2018].
- [5] SATELLINE-M3-TR8 SATELLINE-M3-TR9 TRANSCEIVER MODULES INTEGRATION GUIDE Version 1.7. (n.d.). Retrieved December 9, 2018, from  
<https://fccid.io/MRBSATEL-TA31/User-Manual/Integration-Manual-3580287.html> [Accessed 25 November 2018].
- [6] En.wikipedia.org. (2018). Arduino. [online] Available at:  
<https://en.wikipedia.org/wiki/Arduino> [Accessed 4 May 2018].
- [7] Microchip.com. (2018). Quality | Microchip Technology. [online] Available at:  
<http://www.microchip.com/quality> [Accessed 4 May 2018].
- [8] Mantech.co.za. (2018). Mantech co. [online] Available at:  
<http://www.mantech.co.za/datasheets/products/A000047.pdf> [Accessed 4 May 2018].
- [9] Arduino.cc. (2018). Arduino Nano Manual. [online] Available at:  
<https://www.arduino.cc/en/uploads/Main/ArduinoNanoManual23.pdf> [Accessed 4 May 2018].
- [10] Ni.com. (2018). Ni MyRio. [online] Available at:  
<http://www.ni.com/pdf/manuals/376047c.pdf> [Accessed 4 May 2018].
- [11] The Raspberry Pi. (2018). Exploring Raspberry Pi. [online] Available at:  
<https://cdn.sparkfun.com/datasheets/Dev/RaspberryPi/2020826.pdf> [Accessed 4 May 2018].
- [12] Xbee Series 2 Oem Rf Modules (2018). XBee™ Series 2 OEM RF Modules. [online] Farnell. Available at: <http://www.farnell.com/datasheets/27606.pdf> [Accessed 4 May 2018].

## **Appendix A: Progress Reports**



# Project Management: Progress Report

## Electrical Engineering Department

EEEN4311: Design Methodology & Project Management (Spring 2018)

**Weeks 1-6**

**Progress Report No. 1**

**Date: March 4, 2018**

<b>Project Title</b>	<b>NOMAD ROVER</b>	
<b>TEAM</b>	<b>Abdullah Abdul Kareem (AK)</b>	<b>Faiez Ahmed Feroze (FF)</b>
	<b>Moath Mishal AlSoqair (MS)</b>	<b>Saeed AlMutairi (SM)</b>

<b>Tasks</b>	<b>Description (brief but clear)</b>	<b>% Completion</b>
<b>Completed Weeks 1 to 6</b>	<b>1</b> Background research and reading about <b>Rovers &amp; Robotic Hand</b>	<b>100%</b>
	<b>2</b> Search for components and resources within PMU and local market	
	<b>3</b> Prepared project proposal and project plan	
	<b>4</b> Design for subsystem 1 ( <b>Rover Base</b> )	
	<b>5</b> Design of subsystem 2 ( <b>Rover Controller</b> ).	
	<b>6</b> Search for circuits and components to implement subsystem 1 ( <b>Rover Base</b> ).	
<b>In Progress Weeks 7 &amp; 8</b>	<b>7</b> Acquire Components for SS1 and start assembling	<b>50%</b>
	<b>8</b> Testing SS1 first prototype, and improving	<b>0%</b>
	<b>9</b> Acquire Components for SS2 and start assembling	<b>25%</b>
	<b>10</b> Testing SS2 first prototype, and improving	<b>0%</b>
	<b>11</b> Final prototype for SS1 and SS2 and combining	<b>0%</b>
<b>Comments</b>	Delays will only occur for arrival of products, which are purchased online. So we have decided to order products beforehand, so they can arrive as per the plan.	
<b>Deadlines</b>	<b>ProgRpt1 (Wks 1- 6) Due March 4</b>	<b>ProgRpt3 (Wks 7, 10) Due April 1<sup>st</sup></b>
	<b>ProgRpt2 (Wks 7, 8) Due March 18</b>	<b>ProgRpt4 (Wks 11, 12) Due April 15</b>
<b>File naming: Submit via email &amp; copy all team members &amp; Advisor: e.g. DM_ProgRpt1_Nomad_Rover</b>		



# Project Management: Progress Report

## Electrical Engineering Department

EEEN4311: Design Methodology & Project Management (Spring 2018)

**Weeks 7-8**

**Progress Report No. 2**

**Date: March 18, 2018**

<b>Project Title</b>	<b>NOMAD ROVER</b>	
<b>TEAM</b>	<b>Abdullah Abdul Kareem (AK)</b>	<b>Faiez Ahmed Feroze (FF)</b>
	<b>Moath Mishal AlSoqair (MS)</b>	<b>Saeed AlMutairi (SM)</b>

<b>Tasks</b>	<b>Description (brief but clear)</b>	<b>% Completion</b>
<b>Completed Weeks 1 to 6</b>	1 Background research and reading about <b>Rovers &amp; Robotic Hand</b>	<b>100%</b>
	2 Search for components and resources within PMU and local market	
	3 Prepared project proposal and project plan	
	4 Design for subsystem 1 ( <b>Rover Base</b> )	
	5 Design of subsystem 2 ( <b>Rover Controller</b> ).	
	6 Search for circuits and components to implement subsystem 1 ( <b>Rover Base</b> ).	
<b>In Progress Weeks 7 &amp; 8</b>	7 Acquire Components for SS1 and start assembling	<b>75%</b>
	8 Testing SS1 first prototype, and improving	<b>50%</b>
	9 Acquire Components for SS2 and start assembling	<b>100%</b>
	10 Testing SS2 first prototype, and improving	<b>25%</b>
	11 Final prototype for SS1 and SS2 and combining	<b>0%</b>
<b>Comments</b>	Delays will only occur for arrival of products, which are purchased online. So we have decided to order products beforehand, so they can arrive as per the plan.	
<b>Deadlines</b>	<b>ProgRpt1 (Wks 1- 6) Due March 4</b>	<b>ProgRpt3 (Wks 7, 10) Due April 1<sup>st</sup></b>
	<b>ProgRpt2 (Wks 7, 8) Due March 18</b>	<b>ProgRpt4 (Wks 11, 12) Due April 15</b>
<b>File naming: Submit via email &amp; copy all team members &amp; Advisor: e.g. DM_ProgRpt1_Nomad_Rover</b>		



# Project Management: Progress Report

## Electrical Engineering Department

EEEN4311: Design Methodology & Project Management (Spring 2018)

**Weeks 9-10**

**Progress Report No. 3**

**Date: April 1<sup>st</sup>, 2018**

<b>Project Title</b>	<b>NOMAD ROVER</b>	
<b>TEAM</b>	<b>Abdullah Abdul Kareem (AK)</b>	<b>Faiez Ahmed Feroze (FF)</b>
	<b>Moath Mishal AlSoqair (MS)</b>	<b>Saeed AlMutairi (SM)</b>

<b>Tasks</b>	<b>Description (brief but clear)</b>	<b>% Completion</b>
<b>Completed Weeks 1 to 6</b>	1 Background research and reading about <b>Rovers &amp; Robotic Hand</b>	<b>100%</b>
	2 Search for components and resources within PMU and local market	
	3 Prepared project proposal and project plan	
	4 Design for subsystem 1 ( <b>Rover Base</b> )	
	5 Design of subsystem 2 ( <b>Rover Controller</b> ).	
	6 Search for circuits and components to implement subsystem 1 ( <b>Rover Base</b> ).	
<b>In Progress Weeks 7 &amp; 8</b>	7 Acquire Components for SS1 and start assembling	<b>75%</b>
	8 Testing SS1 first prototype, and improving	<b>75%</b>
	9 Acquire Components for SS2 and start assembling	<b>100%</b>
	10 Testing SS2 first prototype, and improving	<b>50%</b>
	11 Final prototype for SS1 and SS2 and combining	<b>0%</b>
<b>Comments</b>	Delays will only occur for arrival of products, which are purchased online. So we have decided to order products beforehand, so they can arrive as per the plan.	
<b>Deadlines</b>	<b>ProgRpt1 (Wks 1- 6) Due March 4</b>	<b>ProgRpt3 (Wks 7, 10) Due April 1<sup>st</sup></b>
	<b>ProgRpt2 (Wks 7, 8) Due March 18</b>	<b>ProgRpt4 (Wks 11, 12) Due April 15</b>
<b>File naming: Submit via email &amp; copy all team members &amp; Advisor: e.g. DM_ProgRpt1_Nomad_Rover</b>		



# Project Management: Progress Report

## Electrical Engineering Department

EEEN4311: Design Methodology & Project Management (Spring 2018)

**Weeks 11-12**

**Progress Report No. 4**

**Date: April 15, 2018**

<b>Project Title</b>	<b>NOMAD ROVER</b>	
<b>TEAM</b>	<b>Abdullah Abdul Kareem (AK)</b>	<b>Faiez Ahmed Feroze (FF)</b>
	<b>Moath Mishal AlSoqair (MS)</b>	<b>Saeed AlMutairi (SM)</b>

<b>Tasks</b>	<b>Description (brief but clear)</b>	<b>% Completion</b>
<b>Completed Weeks 1 to 6</b>	<b>1</b> Background research and reading about <b>Rovers &amp; Robotic Hand</b>	<b>100%</b>
	<b>2</b> Search for components and resources within PMU and local market	
	<b>3</b> Prepared project proposal and project plan	
	<b>4</b> Design for subsystem 1 ( <b>Rover Base</b> )	
	<b>5</b> Design of subsystem 2 ( <b>Rover Controller</b> ).	
	<b>6</b> Search for circuits and components to implement subsystem 1 ( <b>Rover Base</b> ).	
<b>In Progress Weeks 7 &amp; 8</b>	<b>7</b> Acquire Components for SS1 and start assembling	<b>90%</b>
	<b>8</b> Testing SS1 first prototype, and improving	<b>75%</b>
	<b>9</b> Acquire Components for SS2 and start assembling	<b>100%</b>
	<b>10</b> Testing SS2 first prototype, and improving	<b>75%</b>
	<b>11</b> Final prototype for SS1 and SS2 and combining	<b>25%</b>
<b>Comments</b>	Delays will only occur for arrival of products, which are purchased online. So we have decided to order products beforehand, so they can arrive as per the plan.	
<b>Deadlines</b>	<b>ProgRpt1 (Wks 1- 6) Due March 4</b>	<b>ProgRpt3 (Wks 7, 10) Due April 1<sup>st</sup></b>
	<b>ProgRpt2 (Wks 7, 8) Due March 18</b>	<b>ProgRpt4 (Wks 11, 12) Due April 15</b>
<b>File naming: Submit via email &amp; copy all team members &amp; Advisor: e.g. DM_ProgRpt1_Nomad_Rover</b>		



# Project Management: Progress Report

## Electrical Engineering Department

EEEN4311: Design Methodology & Project Management (Fall 2018-19)

**Weeks 1-6**

**Progress Report No. 1**

**Date: Oct 11, 2018**

<b>Project Title</b>	<b>NOMAD ROVER</b>		
<b>TEAM</b>	<b>ABDULLAH A. KAREEM (AK)</b>	<b>MOATH M. ALSOQAIR (MS)</b>	
	<b>FAIEZ A. FEROZE (FF)</b>	<b>SAEED M. ALMUTAIRI (SM)</b>	

<b>Tasks</b>	<b>Description (brief but clear)</b>	<b>Date</b>	<b>% Completion</b>
<b>Completed Weeks 1 to 6</b>	<b>1</b>	Finished design of subsystem 1 ( <b>ROBOTIC ARM</b> )	Sep 17 <b>100%</b>
	<b>2</b>	Tested subsystem 1 and made necessary tuning	Oct 15 <b>75%</b>
	<b>3</b>	Finished design of subsystem 2 ( <b>ROBOTIC ARM CONTROLLER</b> )	Sep 17 <b>100%</b>
	<b>4</b>	Tested subsystem 2 and made necessary tuning	Oct 15 <b>75%</b>
<b>In Progress Weeks 7 &amp; 8</b>	<b>5</b>	Integrate all subsystems	Oct 22 <b>0%</b>
	<b>6</b>	Making improvements	Oct 29 <b>20%</b>
	<b>7</b>	Final Presentation Preparation	Nov 26 <b>10%</b>
	<b>8</b>	Final Demo and Video Preparation	Nov 05 <b>5%</b>
	<b>9</b>	Final Report Preparation	Nov 19 <b>20%</b>

<b>Comments</b>	With every member working in coordination, we were able to complete all tasks as per schedule. And we will be proceeding as per the plan.		
-----------------	-------------------------------------------------------------------------------------------------------------------------------------------	--	--

<b>Deadlines Thu</b>	<b>ProgRpt1 (Wks 1- 6) Th Oct 11</b>		<b>ProgRpt3 (Wks 9, 10) Th Nov 8</b>
	<b>ProgRpt2 (Wks 7, 8) Th Oct 25</b>		<b>ProgRpt4 (Wks 11, 12) Th Nov 22</b>

**File naming:** Submit via email & **copy all team** members & **Advisor:** e.g.  
**DM\_ProgRpt1\_Nomad\_Rover**



# Project Management: Progress Report

## Electrical Engineering Department

EEEN4311: Design Methodology & Project Management (Fall 2018-19)

**Weeks 7-8**

**Progress Report No. 2**

**Date: Oct 25, 2018**

<b>Project Title</b>	<b>NOMAD ROVER</b>		
<b>TEAM</b>	<b>ABDULLAH A. KAREEM (AK)</b>	<b>MOATH M. ALSOQAIR (MS)</b>	
	<b>FAIEZ A. FEROZE (FF)</b>	<b>SAEED M. ALMUTAIRI (SM)</b>	

<b>Tasks</b>	<b>Description (brief but clear)</b>	<b>Date</b>	<b>% Completion</b>
<b>Completed Weeks 1 to 6</b>	<b>1</b>	Finished design of subsystem 1 ( <b>ROBOTIC ARM</b> )	Sep 17 <b>100%</b>
	<b>2</b>	Tested subsystem 1 and made necessary tuning	Oct 15 <b>75%</b>
	<b>3</b>	Finished design of subsystem 2 ( <b>ROBOTIC ARM CONTROLLER</b> )	Sep 17 <b>100%</b>
	<b>4</b>	Tested subsystem 2 and made necessary tuning	Oct 15 <b>75%</b>
<b>In Progress Weeks 7 &amp; 8</b>	<b>5</b>	Integrate all subsystems	Oct 22 <b>50%</b>
	<b>6</b>	Making improvements	Oct 29 <b>20%</b>
	<b>7</b>	Final Presentation Preparation	Nov 26 <b>10%</b>
	<b>8</b>	Final Demo and Video Preparation	Nov 05 <b>20%</b>
	<b>9</b>	Final Report Preparation	Nov 19 <b>50%</b>

<b>Comments</b>	With every member working in coordination, we were able to complete all tasks as per schedule. And we will be proceeding as per the plan.		
-----------------	-------------------------------------------------------------------------------------------------------------------------------------------	--	--

<b>Deadlines Thu</b>	<b>ProgRpt1 (Wks 1- 6) Th Oct 11</b>		<b>ProgRpt3 (Wks 9, 10) Th Nov 8</b>
	<b>ProgRpt2 (Wks 7, 8) Th Oct 25</b>		<b>ProgRpt4 (Wks 11, 12) Th Nov 22</b>

**File naming:** Submit via email & copy all team members & **Advisor:** e.g.  
**DM\_ProgRpt1\_Nomad\_Rover**



# Project Management: Progress Report

## Electrical Engineering Department

EEEN4311: Design Methodology & Project Management (Fall 2018-19)

**Weeks 9-10**

**Progress Report No. 3**

**Date: Nov 9, 2018**

<b>Project Title</b>	<b>NOMAD ROVER</b>	
<b>TEAM</b>	<b>ABDULLAH A. KAREEM (AK)</b>	<b>MOATH M. ALSOQAIR (MS)</b>
	<b>FAIEZ A. FEROZE (FF)</b>	<b>SAEED M. ALMUTAIRI (SM)</b>

<b>Tasks</b>	<b>Description (brief but clear)</b>	<b>Date</b>	<b>% Completion</b>
<b>Completed Weeks 1 to 6</b>	<b>1</b>	Finished design of subsystem 1 ( <b>ROBOTIC ARM</b> )	Sep 17 <b>100%</b>
	<b>2</b>	Tested subsystem 1 and made necessary tuning	Oct 15 <b>100%</b>
	<b>3</b>	Finished design of subsystem 2 ( <b>ROBOTIC ARM CONTROLLER</b> )	Sep 17 <b>100%</b>
	<b>4</b>	Tested subsystem 2 and made necessary tuning	Nov 15 <b>75%</b>
<b>In Progress Weeks 7 &amp; 8</b>	<b>5</b>	Integrate all subsystems	Nov 20 <b>50%</b>
	<b>6</b>	Making improvements	Nov 29 <b>30%</b>
	<b>7</b>	Final Presentation Preparation	Dec 10 <b>50%</b>
	<b>8</b>	Final Demo and Video Preparation	Dec 10 <b>5%</b>
	<b>9</b>	Final Report Preparation	Nov 19 <b>30%</b>

**Comments**

With every member working in coordination, we were able to complete all tasks as per schedule. And we will be proceeding as per the plan.

**Deadlines Thu**

**ProgRpt1 (Wks 1- 6) Th Oct 11**

**ProgRpt3 (Wks 9, 10) Th Nov 8**

**ProgRpt2 (Wks 7, 8) Th Oct 25**

**ProgRpt4 (Wks 11, 12) Th Nov 22**

**File naming:** Submit via email & **copy all team** members & **Advisor:** e.g.  
**DM\_ProgRpt1\_Nomad Rover**



# Project Management: Progress Report

## Electrical Engineering Department

EEEN4311: Design Methodology & Project Management (Fall 2018-19)

**Weeks 9-10**

**Progress Report No. 4**

**Date: Nov 22, 2018**

<b>Project Title</b>	<b>NOMAD ROVER</b>		
<b>TEAM</b>	<b>ABDULLAH A. KAREEM (AK)</b>	<b>MOATH M. ALSOQAIR (MS)</b>	
	<b>FAIEZ A. FEROZE (FF)</b>	<b>SAEED M. ALMUTAIRI (SM)</b>	

<b>Tasks</b>	<b>Description (brief but clear)</b>	<b>Date</b>	<b>% Completion</b>	
<b>Completed Weeks 1 to 6</b>	<b>1</b>	Finished design of subsystem 1 ( <b>ROBOTIC ARM</b> )	Sep 17	<b>100%</b>
	<b>2</b>	Tested subsystem 1 and made necessary tuning	Oct 15	<b>100%</b>
	<b>3</b>	Finished design of subsystem 2 ( <b>ROBOTIC ARM CONTROLLER</b> )	Sep 17	<b>100%</b>
	<b>4</b>	Tested subsystem 2 and made necessary tuning	Nov 15	<b>100%</b>
<b>In Progress Weeks 7 &amp; 8</b>	<b>5</b>	Integrate all subsystems	Nov 20	<b>100%</b>
	<b>6</b>	Making improvements	Nov 29	<b>100%</b>
	<b>7</b>	Final Presentation Preparation	Dec 10	<b>100%</b>
	<b>8</b>	Final Demo and Video Preparation	Dec 10	<b>100%</b>
	<b>9</b>	Final Report Preparation	Nov 19	<b>100%</b>

<b>Comments</b>	With every member working in coordination, we were able to complete all tasks as per schedule. And we will be proceeding as per the plan.		
-----------------	-------------------------------------------------------------------------------------------------------------------------------------------	--	--

<b>Deadlines Thu</b>	<b>ProgRpt1 (Wks 1- 6) Th Oct 11</b>		<b>ProgRpt3 (Wks 9, 10) Th Nov 8</b>
	<b>ProgRpt2 (Wks 7, 8) Th Oct 25</b>		<b>ProgRpt4 (Wks 11, 12) Th Nov 22</b>

**File naming:** Submit via email & copy all team members & **Advisor:** e.g. DM\_ProgRpt1\_Nomad\_Rover

## Appendix B: Bill of Materials

*Table B.1 Final Cost*

<b>S.No.</b>	<b>Description</b>	<b>Quantity</b>	<b>Unit Cost (SAR)</b>	<b>Total Cost (SAR)</b>
1	ATMEGA 328 + Programming Kit	6	200	1200
2	Stainless Steel Car Chassis (with motor)	1	1200	1200
3	Camera Kit	2	400	800
4	HC-12	5	35	175
5	4 DOF Arm and hand	1	1300	1300
6	Assorted LED kits	2	100	200
7	Controller (Joystick)	1	50	50
8	Battery System (Li-Ion- LiPo)	5	60	300
9	Miscellaneous	-	-	1000
10	L293N Motor driver	1	75	258
11	Breadboard, Jumpers	1	200	200
12	Soldering (Solder+ tools)	1	150	150
13	Tool Set	1	75	75
<b>TOTAL</b>				<b>6,908</b>

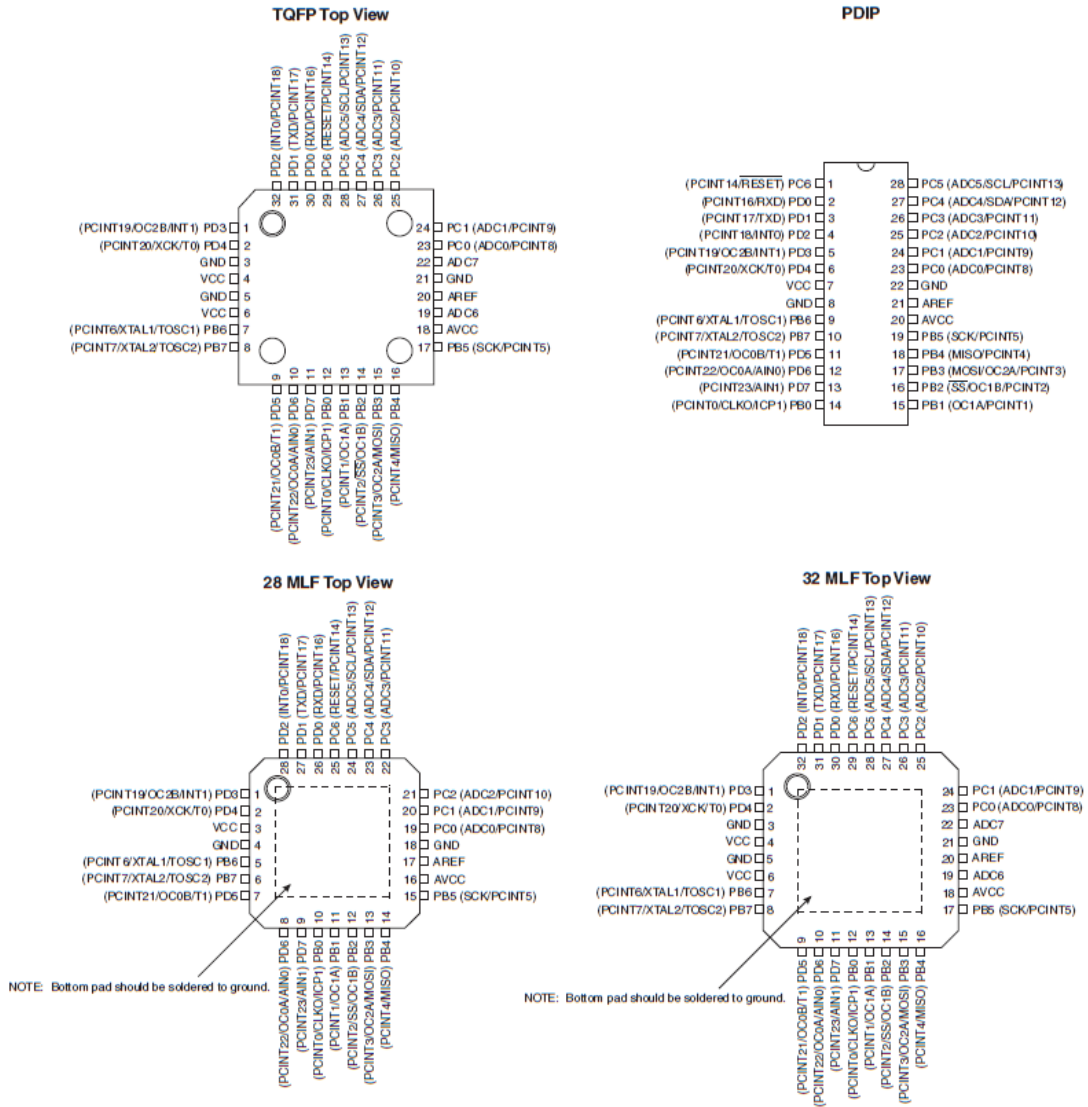
## **Appendix C: Datasheets**

# ATMEGA 328 - DATASHEET

## ATmega48PA/88PA/168PA/328P

### 1. Pin Configurations

Figure 1-1. Pinout ATmega48PA/88PA/168PA/328P



## 1.1 Pin Descriptions

### 1.1.1 VCC

Digital supply voltage.

### 1.1.2 GND

Ground.

### 1.1.3 Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB7..6 is used as TOSC2..1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

The various special features of Port B are elaborated in ["Alternate Functions of Port B" on page 76](#) and ["System Clock and Clock Options" on page 26](#).

### 1.1.4 Port C (PC5:0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5..0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

### 1.1.5 PC6/RESET

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in [Table 28-3 on page 308](#). Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated in ["Alternate Functions of Port C" on page 79](#).

### 1.1.6 Port D (PD7:0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

## ATmega48PA/88PA/168PA/328P

The various special features of Port D are elaborated in "[Alternate Functions of Port D](#)" on page 82.

### 1.1.7 $AV_{CC}$

$AV_{CC}$  is the supply voltage pin for the A/D Converter, PC3:0, and ADC7:6. It should be externally connected to  $V_{CC}$ , even if the ADC is not used. If the ADC is used, it should be connected to  $V_{CC}$  through a low-pass filter. Note that PC6..4 use digital supply voltage,  $V_{CC}$ .

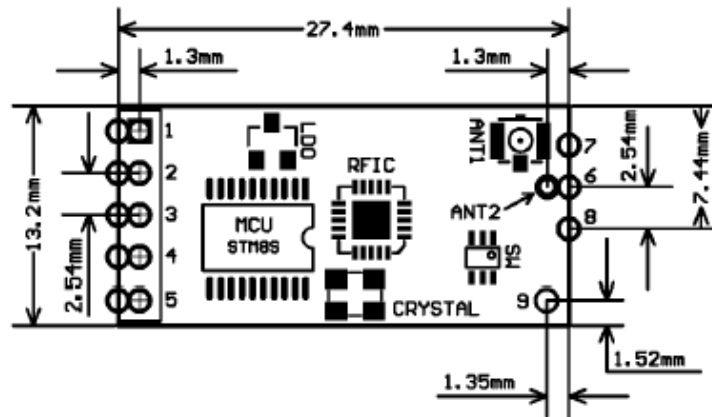
### 1.1.8 AREF

AREF is the analog reference pin for the A/D Converter.

### 1.1.9 ADC7:6 (TQFP and QFNMLF Package Only)

In the TQFP and QFNMLF package, ADC7:6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

### Product Dimensions



### Definition of Pins

The HC-12 module can be patch soldered, or have a 2.54mm-spacing pin header attached and directly inserted onto the user's PCB. The module has nine pins in total, and one RF antenna socket (ANT1), with definitions as shown in the table below:

Pin	Definition	I/O direction	Notes
1	Vcc		Power supply input, DC3.2V-5.5V, with load capacity not less than 200mA. Note: if the module is working in the transmitting state for an extended time, it is suggested that a 1N4007 diode be connected in series if the supply voltage is greater than 4.5V, so as to avoid overheating the onboard LDO regulator
2	GND		Common ground
3	RxD	Input (weak pullup)	UART data input, TTL level. 1k resistor connected in series inside the module
4	TxD	Output	UART data output, TTL level. 1k resistor connected in series inside the module
5	SET	Input (10k pullup)	Parameter setting control pin, active low level. 1k resistor connected in series inside the module
6	ANT	Input/Output	433MHz antenna pin
7	GND		Common ground
8	GND		Common ground
9	NC		No connection, used in mechanical fixing, compatible with HC-11 module pin position
ANT1	ANT	Input/Output	IPEX20279-001E-03 antenna socket
ANT2	ANT	Input/Output	433MHz spring antenna solder eyelet

Pins 1-6 each have two bonding pads, with the outer half-hole bonding pads intended for patch soldering. When the inner bonding pad ANT2 of Pin 6 is used for connection, the spring antenna can be soldered here by hand. The inner round-hole bonding pads of Pins 1-5 may then be used to solder a 2.54mm-spacing pin header that can be plugged into a PCB socket.

## **Appendix D: Program Codes**

## ROBOTIC HAND RECEIVER

PAGE 1/3	PAGE 2/3	PAGE 3/3
<pre> 1.  #include &lt;SoftPWM.h&gt; 2.  #include    &lt;SoftwareSerial.h&gt; 3.  #include &lt;Servo.h&gt; 4.  const int motor1 = 12    ; 5.  const int motor2 =    11; 6.  const int motor3 = A4    ; 7.  const int motor4 =    A5; 8.  const int en1 = 13 ; 9.  const int en2 = A3;  10. Servo finger1; //    create servo object to    control a servo 11. Servo finger2; //    create servo object to    control a servo 12. Servo finger3; //    create servo object to    control a servo 13. Servo finger4; // 14. to control a servo  15. SoftwareSerial    mySerial(A1, A0);    // assign RX, TX to GPIO    pins 16. char serialStr[110] =    "";    //define max limit for    char array to store    incoming data 17. char c = 0;    //define c as char to    store one byte (single    data) 18. bool data = false;    //define conditional    operator data as bool to    check condition  19. char x_axis[2]; 20. char y_axis[2];  21. char fingers[6][13]; 22. int finger02; 23. String finger01; 24. int val; 25. int val1; 26. int val2; 27. int val3; 28. int val4; 29. int val5;  30. String finger11; 31. String finger21; 32. String finger31; 33. String finger41; 34. String finger51;  35. String x_axis1; 36. String y_axis1;  37. void setup() { 38. // put your setup    code here, to run once: 39. Serial.begin(115200); </pre>	<pre> 52. void loop() { 53. // put your main code    here, to run repeatedly: 54. //scanf(serialStr,    "{%*[^\n]T%[^\n]!",    &amp;finger1); 55. while (1) 56. { 57. if    (Serial.available()) 58. { 59. while (c != '&amp;')    //execute while loop until    c reads as closing curly    bracket '}' 60. { 61. if    (Serial.available()) 62. { 63. c=(Serial.read());    //Read single byte of data    (i.e. character)from soft    serial and store it in c    char 64. if (c == '\0') 65. serialStr[arrIndex] =    c; //transfer    character stored in c to    character array    'serialStr' 66. arrIndex++ ;    //increment this counter 67. } 68. } 69. serialStr[arrIndex] =    '\0'; 70. , &amp;fingers4,    &amp;fingers5, &amp;x_axis,    &amp;y_axis);  71. finger11 =    String(fingers1); 72. finger21 =    String(fingers4); 73. finger41 =    String(fingers4); 74. finger51 =    String(fingers5);  75. x_axis1 =    String(x_axis); 76. mySerial.print("x_axi    s1:"+ x_axis1); 77. y_axis1 =    String(y_axis); 78. mySerial.print("y_axi    s1:"+ y_axis1);  79. int finger02 =    finger01.toInt(); 80. int finger12 =    finger11.toInt(); 81. int finger22 =    finger21.toInt(); 82. int finger32 =    finger31.toInt(); 83. int finger42 =    finger41.toInt(); 84. int finger52 =    finger51.toInt(); 85. val = finger02; </pre>	<pre> 101. if(x_axis1=="0") 102. { 103. horizontal_motor_off()    ; 104. }  105. else if (x_axis1=="1") 106. { 107. horizontal_motor(1,200    ); 108. }  109. else if (x_axis1=="2") 110. { 111. horizontal_motor(0,200    ); 112. } 113. if(y_axis1=="0") 114. { 115. vertical_motor_off(); 116. } 117. else if (y_axis1=="1") 118. { 119. vertical_motor(1,200); 120. } 121. else if (y_axis1=="2") 122. { 123. vertical_motor(0,200); 124. y_axis1=="0"; 125. }  126. { 127. } 128. } 129. { 130. finger3.write(val3); 131. mySerial.print("new    val3:"); 132. mySerial.println(val3)    ; 133. }  134. { 135. finger4.write(val4); 136. mySerial.print("new    val4:"); 137. mySerial.println(val4)    ; 138. }  139. { 140. finger5.write(val5); 141. mySerial.print("new    val5:"); 142. mySerial.println(val5)    ; 143. } 144. } 145. } 146. }  147. void    horizontal_motor(int    direction1 , int speed1) 148. { 149. if (direction1 == 0) 150. { 151. mySerial.print("direct    ion clockwise"); 152. digitalWrite(motor1,    HIGH); </pre>

<pre> 40.   mySerial.begin(115200 ); //initialize software serial communication 41.   SoftPWMBegin();  42.   SoftPWMSet(en1, 0); 43.   SoftPWMSet(en2, 0); 44.   SoftPWMSetFadeTime(en 1, 1000, 1000); 45.   SoftPWMSetFadeTime(en 2, 1000, 1000); 46.   finger1.attach(10); // attaches the servo on pin 9 to the servo object 47.   finger2.attach(9); // attaches the servo on pin 9 to the servo object 48.   finger3.attach(6); // attaches the servo on pin 9 to the servo object 49.   finger4.attach(5); // attaches the servo on pin 9 to the servo object 50.   finger5.attach(3); // attaches the servo on pin 9 to the servo object 51.   } </pre>	<pre> 86.   val1 = map(finger12, 2000, 2100, 50, 160); //thum 87.   val2 = map(finger22, 1700, 1110, 130, 10); //index finger 88.   val3 = map(finger32, 2100, 1000, 190, 30); //middle finger 89.   val4 = map(finger42, 1170, 1780, 30, 160); //third finger 90.   val5 = map(finger52, 1210, 2100, 170, 60); //little finger  91.   mySerial.print("val1: "); 92.   mySerial.println(val1 ); 93.   mySerial.print("val2: "); 94.   mySerial.println(val2 ); 95.   mySerial.print("val3: "); 96.   mySerial.println(val3 ); 97.   mySerial.print("val4: "); 98.   mySerial.println(val4 ); 99.   mySerial.print("val5: "); 100.  mySerial.println(val5 ); </pre>	<pre> 153.  digitalWrite(motor2, LOW); 154.  } 155.  else if(direction1 == 1) 156.  { 157.  void vertical_motor(int direction1 , int speed1) 158.  { 159.  if (direction1 == 0) 160.  { 161.  mySerial.print("direct ion clockwise"); 162.  digitalWrite(motor3, HIGH); 163.  digitalWrite(motor4, LOW); 164.  } 165.  else if(direction1 == 1) 166.  { 167.  mySerial.print("direct ion anti-clockwise"); 168.  digitalWrite(motor3, LOW); 169.  digitalWrite(motor4, HIGH); 170.  } 171.  SoftPWMSet(en2, speed1); 172.  } 173.  void vertical_motor_off() 174.  { 175.  digitalWrite(motor3, LOW); 176.  digitalWrite(motor4, LOW); 2.  } </pre>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## ROBOTIC ARM CONTROLLER

PAGE 1/2	PAGE 2/2
<pre> 1.  #include &lt;SoftwareSerial.h&gt; 2.  //SoftwareSerial mySerial(9, 10); // RX, TX 3.  char serialStr[110] = ""; //define max limit for char array to store incoming data 4.  char c = 0; //define c as char to store one byte ( 5.  int acc_y; 6.  int led = 5; 7.  String data_to_send; 8.  int x, y, x_val, y_val, last_val, val, stability; 9.  void setup() 10. { 11. Serial.begin(115200); 12. // mySerial.begin(115200); 13. pinMode(led, OUTPUT); 14. digitalWrite(led, LOW); 15. x_val = checkstabiliy(A1); 16. y_val = checkstabiliy(A2); 17. digitalWrite(led, HIGH);  18. } 19. void loop() </pre>	<pre> 33. if (data) 34. { 35. data_to_send = "";  36. acc_x = analogRead(A1); 37. x = acc_x - x_val; 38. if (x &lt; -30) 39. { a. data_to_send = String(serialStr) + "@1\$"; 40. } 41. else if (x &gt; 30) 42. { a. data_to_send = String(serialStr) + "@2\$"; 43. } 44. else 45. { a. data_to_send = String(serialStr) + "@0\$"; 46. } a. "2&amp;"; 47. } 48. else 49. { a. data_to_send = data_to_send + "0&amp;"; </pre>

<pre> 20. { 21. data_to_ 22. '}', 23. { 24. if (Serial.available()) 25. {     a. c = (Serial.read());         //Read single byte of         data (i.e. character)from         soft serial and store it         in c char     b. if (c == '\0')     c. serialStr[arrIndex] = c;         //transfer character         stored in c to character         array 'serialStr'     d. arrIndex++ ;         //increment this counter 26. } 27. } 28. serialStr[arrIndex] = '\0'; 29. arrIndex = 0; 30. data = true; 31. c = 0;  32. } </pre>	<pre> 50. }  51. Serial.println(data_to_send); 52. } 53. ); 54. Serial.println(val); 55. delay(100); 56. stability++; 57. if ((val - last_val) &lt; -5)    ((val -     last_val) &gt; 5)) 58. { 59. stability = 0; 60. last_val = val; 61. } 62. }  63. stability = 0; 64. return val; 65. } </pre>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## ROVER CONTROLLER

### PAGE 1/2

```

1. const int data0=8;
2. const int data1=7;
3. const int x_axis=A3;
4. const int y_axis=A4;
5. int x_value, y_value;

6. void setup() {
7. // put your setup code here, to run once:
8. Serial.begin(9600);
9. pinMode(x_axis, INPUT);
10. pinMode(y_axis, INPUT);
11. pinMode(data0, OUTPUT);
12. pinMode(data1, OUTPUT);
13. }

14. void loop() {
15. // put your main code here, to run repeatedly:
16. Serial.println(x_value);
17. Serial.print("y:");
18. Serial.println(y_value);
19. if(x_value>400)
20. {
21. send_data(1,0,1,1);

22. }
23. else if(x_value<100)
24. {

25. send_data(0,1,1,1);

26. }
27. else if(y_value<100)
28. {

```

```

29. send_data(1,1,1,0); // left x=0
30. }
31. else if(y_value>900)
32. {
33. send_data(1,1,0,1); // right x=1023
34. }
35. else
36. {
37. send_data(1,1,1,1);
38. }

39. }

40. void send_data(bool D0, bool D1, bool D2, bool D3)
41. {
42. digitalWrite(data0,D0);
43. digitalWrite(data1,D1);
44. digitalWrite(data3,D3);
45. }

```

## ROVER RECEIVER

PAGE 1/2	PAGE 2/2
<pre> 1. ///////////////CONSTANT///////////////// 2. int VT_status = 0; 3. int data0 = 0; 4. int data2 = 0; 5. int data3 = 0;  6. ///////////////pins///////////////// 7. const int vt = A0; 8. const int d0 = A4; 9. const int d1 = A3; 10. const int d2 = A2; 11. const int d3 = A1;  12. const int EN1 = 10; 13. const int EN2 = 3; 14. const int IN1 = 9;  15. pinMode(EN1,OUTPUT); 16. pinMode(EN2,OUTPUT); 17. pinMode(IN1,OUTPUT); 18. pinMode(IN2,OUTPUT); 19. pinMode(IN3,OUTPUT); 20. pinMode(IN4,OUTPUT); 21. }  22. void loop() 23. {  24. while (1) 25. {                 VT_status =                 digitalWrite(vt); 26. data0 = digitalRead(d0); 27. data1 = digitalRead(d1); 28. data2 = digitalRead(d2); 29. data3 = digitalRead(d3);                 rf_process(); 30. } 31. } </pre>	<pre> 32. void rf_process() 33. { 34. if ((VT_status == 1) &amp;&amp; (data0 == 0) &amp;&amp;     (data1 == 1) &amp;&amp; (data2 == 1) &amp;&amp; (data3     == 1)) 35. {                 forward();  36. } 37. else if ((VT_status == 1) &amp;&amp; (data0 ==     1) &amp;&amp; (data1 == 0) &amp;&amp; (data2 == 1) &amp;&amp;     (data3 == 1))  38. } 39. else if ((VT_status == 1) &amp;&amp; (data0 ==     1) &amp;&amp; (data1 == 1) &amp;&amp; (data2 == 1) &amp;&amp;     (data3 == 0)) 40. {                 right(); </pre>

# Appendix E: Operation Manual

## RUNNING THE PROTOTYPE

1. In order to run the project prototype, make sure the batteries are connected to the following,
  - a. 9V DC to Rover Controller
  - b. 9V DC to Camera
  - c. 11.1V LiPo to Robotic Arm and Rover base circuit
  - d. Connecting the camera receiver to a wall adapter
2. Once the power supplies are connected, turn on the switches on all the circuits
3. For the rover base, controller will be connected to receiver once the red LEDs stops blinking, and stays on
4. For the robotic hand, controller will be connected to receiver once the blue LEDs is blinking.
5. Now the project is in run mode.

## UPLOADING PROGRAM CODES

1. Copy the code given in APPENDIX D to the Arduino IDE.
2. Double check the PIN numbers of the components in the code and on the Arduino board.
3. Make sure to upload the right code to the right component, example robotic arm controller codes to the robotic arm controller circuit
4. Also remove RX and TX PINS before uploading the code to Arduino to prevent the .avrdude error
5. Once the code is uploaded, reconnect the RX and TX PINS

## LIST EXPECTED OUPUTS

1. The 2-AXIS joystick controller, controls the motion of the car. Pushing it forward, causes the car to move forward. Also, is the same is for reverse and left and right
2. The robotic hand controls both motion of the robotic arm, and the fingers. Moving the user's fingers controls the robotic hand's fingers. Raising or lowering the user's palm raises or lowers the robotic arm, and rotating the user's palm rotates the robotic arm.
3. The camera feed can be obtained by connecting the receiving router to computer/Mac and opening QuickTime.