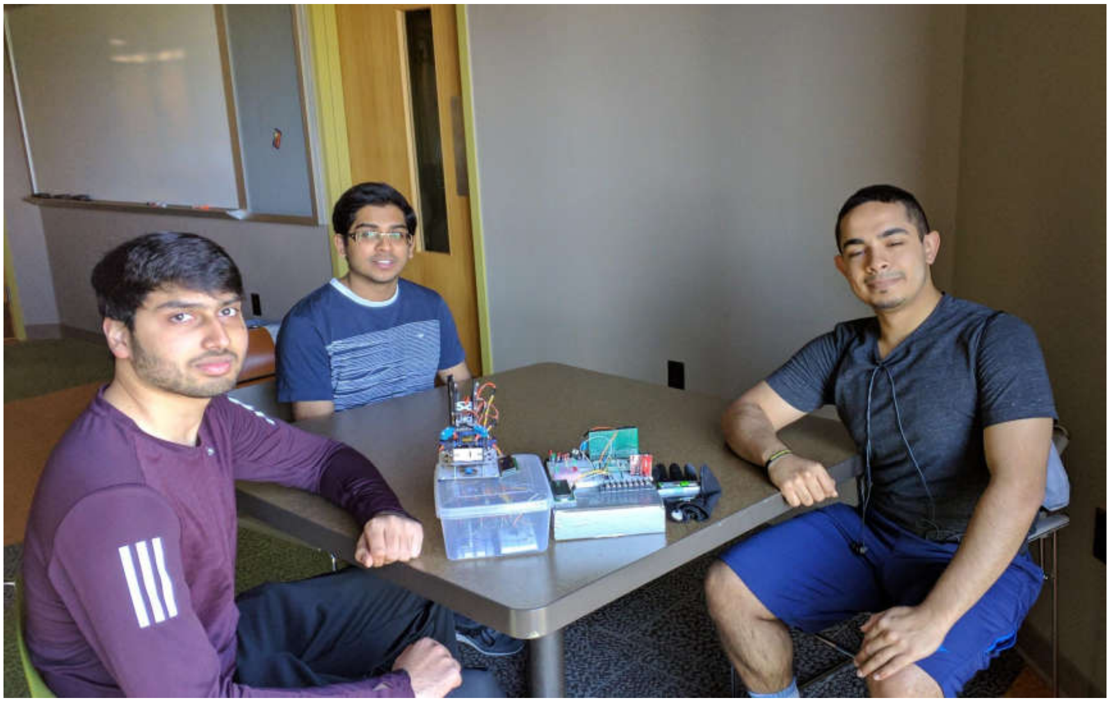
ARM ASSEMBLY



Team Members (left-to-right on picture, above)	*Class No.*	*Lab Div*
Vikrant Satheesh Kumar	****	*
Ajay Gopakumar	****	*
Satyajyoti Nanda	****	*

Introduction

The project ARM Assembly consisted of a robotic arm which was controlled through the operator's gestures. The purpose of the arm is to serve in applications which are too dangerous for the human arm. The robotic arm can be used to handle hazardous material, ordnance or radioactive material etc. The robotic arm is controlled by a series of servo motors, which depend on Pulse Position Modulation (PPM) signals in order to maintain a shaft rotation angle. The motor's PPM signals are driven through the STM32F0 Discovery Board. The board receives the relevant gesture information through a serial wireless interface over Bluetooth. The gestures are captured and transmitted through an MPU-6050 Inertial Measurement Unit (IMU) augmented by an HC-05 Bluetooth interface.

The future vision of this project is to have a well-calibrated arm that follows the operators gestures accurately. Furthermore, for specific applications, the arm can be strengthened to act as a robotic strength augmentation. The team consisted of three members, Ajay Gopakumar, Satyajyoti Nanda, and Vikrant Satheesh Kumar.

Interface Design

BWT61 Bluetooth MPU6050:

The BWT61 Bluetooth module was the heart of the project. The MPU6050 module is a 6-axis Accelerometer and Gyroscope. It has a built-in bluetooth (HC-06) module and also supports wired serial transmission at a baud rate of 115200. The module provides data to the host in 3 data packets (acceleration output, angular velocity output and angular output) of 11 bytes each. The data is transmitted in hexadecimal digits and the first two bytes are used to identify the type of data (acceleration output, angular velocity output and angular output). The conversion values were obtained from the data sheet. The module runs on 3-6V and consumes <10 mA of current. (Citation 3,4,5)

Servo Motors (4) - SG90 MICRO SERVO:

The SG90 micro servos were used to control arm movements. They have a 180 degree rotation angle and weigh about 11 grams. They are able to provide 1.8 kg-cm of torque at 4.8V and take 0.12 sec for changing angles by 60 degrees. They draw a maximum current of 600mA. The servos were controlled using PPM. A 1ms pulse corresponds to the minimum angle of 0 degrees and a 1.5 ms pulse corresponds to a 90 degree angle, finally a 2ms pulse corresponds to the maximum angle of 180 degrees. (Citation 7)

HC-05 Bluetooth Master Module:

The HC-05 is a Serial Bluetooth module for microcontrollers. It works with Serial communication (USART) and is TTL compatible. It has a reported range of <100m. It follows the IEEE 802.15.1 standardized protocol and can operate in Master, Slave or Master/Slave mode. The HC-05 module acts as the master that reads from the gyroscopes HC-06 module. It had to be configured to pair with the HC-06 with the appropriate baud rate using AT command mode. (Citation 5,6)

74HC595 8-bit Shift Register + 16 x 2 LCD:

This IC was utilised to interface a 16 x 2 LCD display to the STM32F0 Discovery Board. The mentioned LCD display uses a parallel 8-bit data bus along with control signal pins. In order to save GPIO pins on the STM32F0, the parallel bus was connected to the shift register outputs. Data bits were shifted into the shift register using the SPI peripheral of the STM32F0.
The LCD was used to display the current rotation position of each of the motors in the robotic arm.(Citation 1,2)

Microcontroller Resource Utilization

The microcontroller peripherals used in the project are described below:

1. Advanced Timer TIM1: This timer's purpose was to provide interrupts to set the correct frequency to update the motor positions by adjusting the pulse widths on their respective data lines. This timer was configured with a pre-scaler of 47999 and an auto-reload of 49. This set the interrupt period at ~50 ms. Furthermore, the update interrupt was enabled. This interrupt period was chosen as it provided a good balance between response time and over-adjustment of motors.
2. General Timer TIM2: This timer's purpose was to provide the control pulses to the motors using Pulse Width Modulation (PWM) mode. The pre-scaler was set to 47 and the auto-reload was set to 9999. This made the period of each PWM cycle ~10 ms. Channels 1-4 were enabled and configured in PWM mode. Their compare registers were altered by the software depending on the desired motor position. No interrupt was enabled for this peripheral, since the PWM controls motors and there is no feedback desired from the motors.
3. Universal Asynchronous Port USART1: This peripheral was used to receive inertial measurements from the operator's glove. A baud rate of 115200 was chosen, along with an 8-bit data + 1-bit stop packet format. Such transmission characteristics were chosen to ensure compatibility with the IMU used on the operator's glove. This USART1 was connected to a Bluetooth receiver module which formed the wireless link between the operator's glove and the microcontroller. The data receipt interrupt was enabled since the microcontroller was to process relevant information from the IMU raw data as soon as a measurement was received by the microcontroller.
4. Serial Peripheral Port SPI2: This peripheral was used to transmit data to the 74HC595 shift-register which in turn was interfaced with a standard 16 x 2 character LCD. This peripheral was configured with a pre-scaler of 256, resulting in a transfer rate of 187.5 kbits/s. The Motorola frame format was chosen. Also, the clock polarity was low and the clock phase was set to 1 edge. This configuration was chosen to ensure accurate data transfer to the shift-register, ultimately ensuring correct functioning of the character LCD. No interrupt was enabled as no data was being received from the LCD itself.

Other significant microcontroller resources used were:

1. System Clock: The system clock was set at the default of 48 MHz for maximum processing power, considering that floating point operations were required to process the raw data from the IMU.
2. NVIC: The interrupt controller was programmed to ensure a sensible priority of interrupts depending on criticality of operation. The SysTick interrupt was given the highest priority for the usage of HAL_Delay(). Next came the USART1 receipt interrupt. This was to ensure receipt of accurate data. Finally, the TIM1 interrupt was placed as the motors could only be adjusted after receiving fresh data from the IMU through USART1.
3. SysTick Timer: This resource was required to provide requisite delays during LCD operation.

Software Narrative

The software driving the microcontroller in this project is organised in a flag-driven format. The overall system flow is described flow:

1. Initialisation: In this phase, all peripherals are initialised and configured to the desired mode. Most of this is done by HAL initialisations. The character LCD is initialised by custom functions. The arm control motors are initialised to known middle positions. The interrupt timer count is started along with the PWM channel counters. Finally, all motor control flags are initialised to reflect lack of movement.

2. Data Acquisition: The USART1 peripheral interrupts the controller as soon as a packet of readings from the IMU is received. As soon as the packet is received, it is decomposed into angular velocity and z-axis acceleration using various mathematical operations specified in the data sheet. Furthermore, various global flags are set, depending on the angular velocities measured by the IMU.

3. Control Motor Adjustment: The TIM1 peripheral interrupts the controller at a predetermined frequency to adjust the control motors. Depending on the values of the flags set during the data acquisition phase, the pulse widths of the motor data lines are adjusted to reflect the gesture broadcasted by the operator's glove. For visual feedback, the current states of the motors are displayed on the character LCD through the SPI2 peripheral.

4. Repetition: These processes are repeated at a frequency close to the TIM1 interrupt frequency (~20 Hz)

5. Virtual Floating Point: This project utilises floating point operations to translate raw data from the IMU into meaningful angular velocities. This requires a floating point unit which is missing on the STM32F0. Therefore, these are emulated using the built-in virtual floating point emulator.

Packaging Design

The project includes three major components: a control box, a gesture glove, and the robotic arm itself.

Control Box: A transparent plastic box with 2 input power ports and 4 output control ports. The 2 input power ports are rated for 110 - 240 VAC and have a rated output of 5 V_{DC} / 2.1 A and 5 V_{DC} / 1A respectively. Both need to be connected for the unit to function correctly. The output control ports are connected to the motor lines on the robotic arm. The control box contains the microcontroller, bluetooth receiver and supporting circuitry. It also includes an LCD display for the motor position status.

Gesture Glove: The gesture glove is used by the operator to control the robotic arm's movements. The gesture glove includes a mounted MPU-6050 IMU and a battery pack(power source). The gyroscope is powered by 4 x NiMH 1.2V AA cells. The gyroscope can also be powered by 1.5 V alkaline cells as long as the combined voltage does not exceed 6 VDC.

Robotic Arm: The robotic arm is a market-fabricated unit that requires self assembly. The robotic arm possesses 4 servo motors which control its motion. Each motor has a separate set of input lines. Each line contains a power, data, and ground input. Each motor is not expected to draw more than 600 mA, based on available datasheets for 9g mini-servo motors.

Summary and Conclusions

Creating a functional gesture controlled robotic arm was a great learning experience for us. We faced several challenges along the way, for instance, acquiring wrong sensor data, illegal sweep angles for motors, low power supply, etc. All of these challenges helped us to learn better and complete the project. We chose the project as we saw a tremendous scope for such a project in areas which might be dangerous for humans. This project can be directly applied in areas such as remote surgeries, handling radioactive waste material, bomb defusal, exoskeletons, etc.

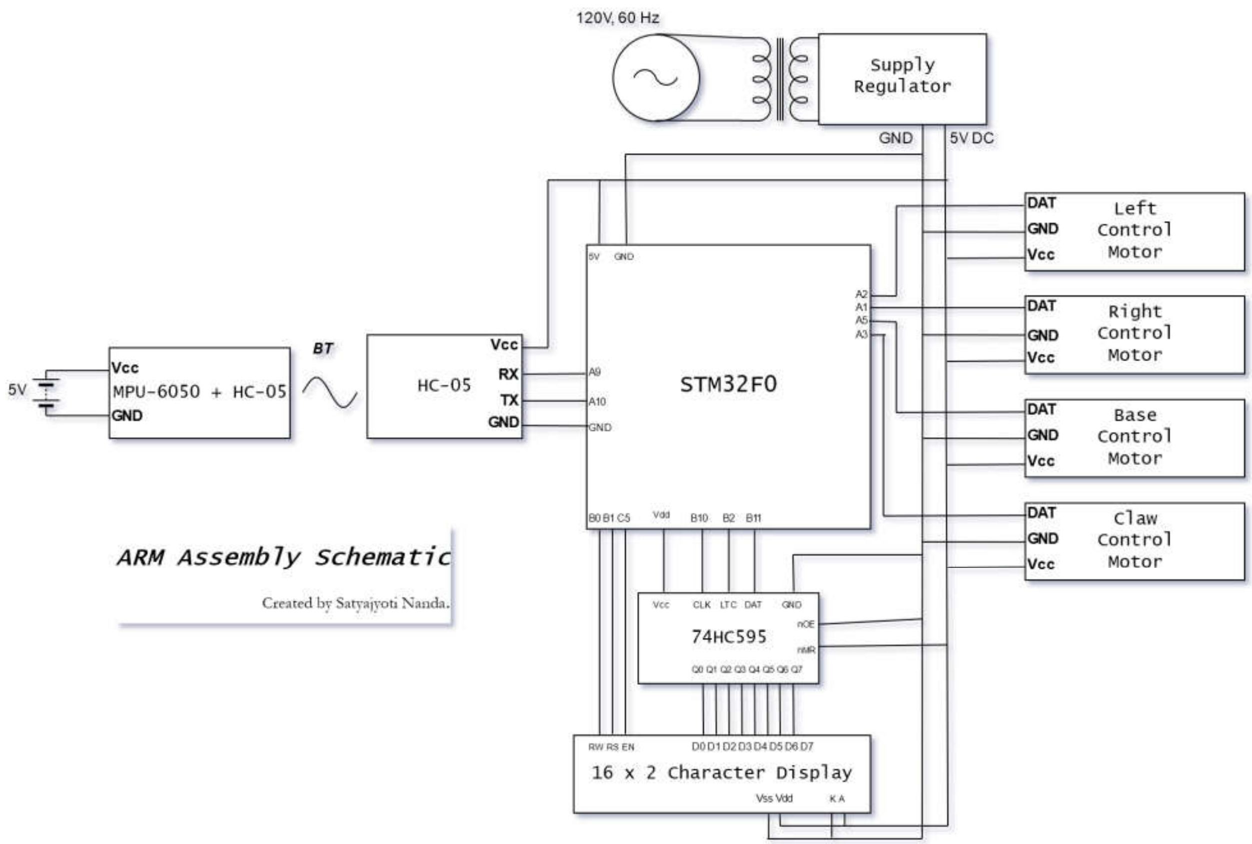
If we had more time and sufficient funding, we could build a more powerful arm which provides more torque which could provide humans with more lifting capability. We would also significantly improve the accuracy of the accelerometer readings to make the mapping between the gesture control and the motors more smoother. Currently, our project requires the user to provide a jerk along the z-axis to open and close the robotic claw. We could replace this jerk action with a button to ensure that the arm's position doesn't change as the claw is opened and closed.

1.0 References

[1] C. Maw and T. I. Nov, "Serial to Parallel Shifting-Out with a 74HC595," [Online]. Available: https://www.arduino.cc/en/Tutorial/ShiftOut. [Accessed: Apr. 20, 2018].
[2] Unknown, "LCD Commands" [Online]. Available: https://mil.ufl.edu/3744/docs/lcdmanual/commands.html. [Accessed: Apr. 20, 2018].
[3] Invensense.com. (2013). *MPU-6000 Datasheet*. [online] Available at: https://www.invensense.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf [Accessed 14 Apr. 2018].
[4] ElecMaster (2013). *JY-61 User Manual*. [online] Available at: https://drive.google.com/file/d/0B8PmY6nhQadKWGNJbmNoRTBtNjA/view?usp=sharing [Accessed 14 Apr. 2018].
[5] Electronicaestudio.com. (2010). *HC-05 module*. [online] Available at: http://www.electronicaestudio.com/docs/istd016A.pdf [Accessed 14 Apr. 2018].
[6] Martyncurrey.com. (2018). *HC-05 with firmware 2.0-20100601 | Martyn Currey*. [online] Available at: http://www.martyncurrey.com/hc-05-with-firmware-2-0-20100601/#AT-Command-Mode [Accessed 15 Apr. 2018].
[7] SG90 servo motor Datasheet. [online] Available at: http://www.ee.ic.ac.uk/pcheung/teaching/DE1_EE/stores/sg90_datasheet.pdf [Accessed 15 Apr. 2018].

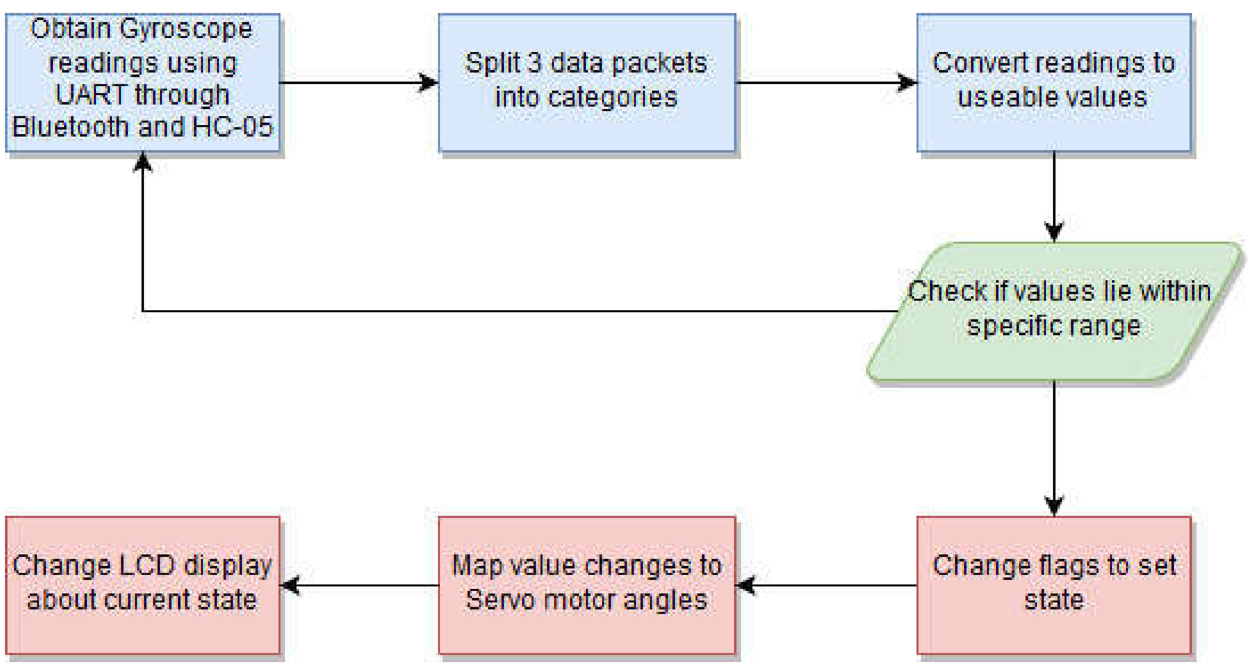
Appendix

Interface Schematic
and
PCB Layout Design



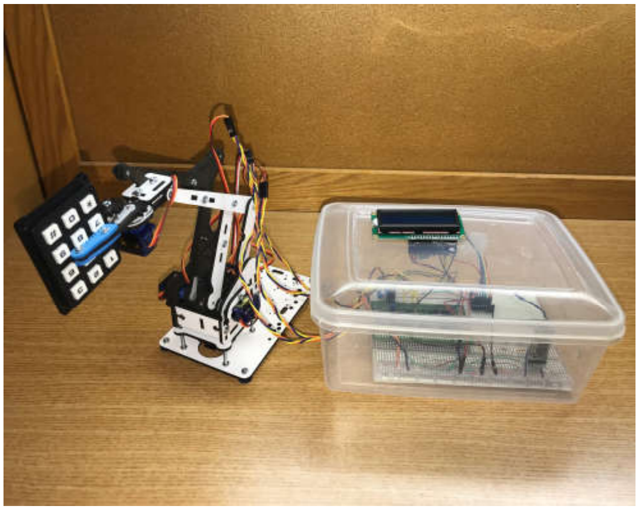
120V, 60 Hz
Supply Regulator
GND
5V DC
5V
GND
STM32F0
A2
A1
A5
A3
A9
A10
GND
DAT
GND
Vcc
Left Control Motor
DAT
GND
Vcc
Right Control Motor
DAT
GND
Vcc
Base Control Motor
DAT
GND
Vcc
Claw Control Motor
5V
Vcc
MPU-6050 + HC-05
GND
BT
HC-05
Vcc
RX
TX
GND
B0 B1 C5
Vdd
B10
B2
B11
Vcc
CLK
LTC
DAT
GND
74HC595
nOE
nMR
Q0 Q1 Q2 Q3 Q4 Q5 Q6 Q7
RW RS EN
D0 D1 D2 D3 D4 D5 D6 D7
16 x 2 Character Display
Vss Vdd
K A
ARM Assembly Schematic
Created by Satyajyoti Nanda.

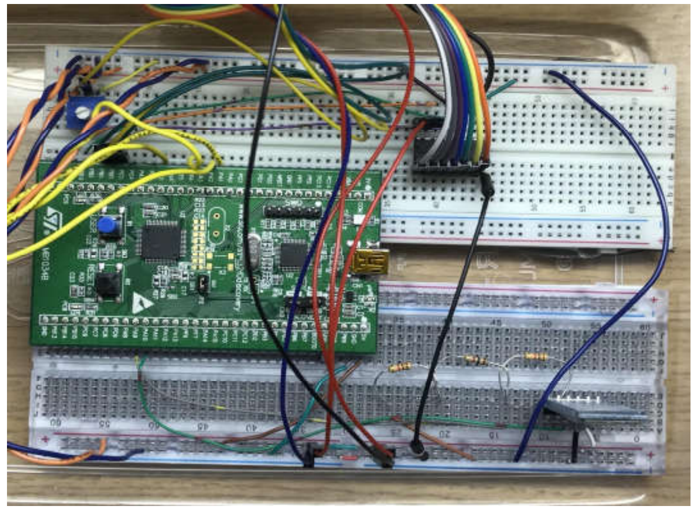
Software Flowcharts


Obtain Gyroscope readings using UART through Bluetooth and HC-05
Split 3 data packets into categories
Convert readings to useable values
Check if values lie within specific range
Change flags to set state
Map value changes to Servo motor angles
Change LCD display about current state

Packaging Design

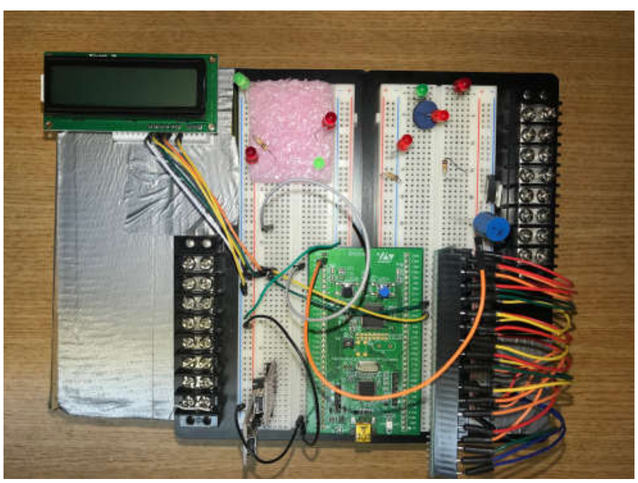
Product Packaging





Robotic ARM and ARM Control Unit

ARM Circuit



Bomb Defusal Unit

Accelerometer Glove