# Man v/s Bot : Hardware Final Report

Mihir Patankar - A53103650
Vignesh Srinivasan - A53103792

**Abstract:**
*Boardgames form part of a multi million industry. Player to player interaction is the essence of increasing demand for board games. However, in the recent years, this industry has lacked innovation and novelty. In this digitally emerging world, technology has led to wide expanse of innovations. However, research[1] shows that it has also led to an alarming increase in the loneliness which in turn has led to increasing introverts and psychologically unmotivated human beings. We introduce a novel low-cost, portable system called 'man vs bot' to improve user interaction that can be used between a human and a robotic arm to play a 2 player board game. The assembly involves a convolution of hardware and software where a smart Artificial Intelligence algorithm will have the ability to predict its moves based on the situation on-board and make a move with the help of the robotic arm.*

**Introduction:**
The current age although considered as an age of increasing innovation and technological advancements, it also has a flip side to it; the  side that we generally turn a blind eye towards. This age is also known as the “age of loneliness”. Researchers have done a number of experiments that portray that the land of opportunities often turns out to be leading to a land of loneliness. As a result, cities like San Francisco, Dublin, London, Sydney feature in the top 10 cities in WHO’s world loneliness index[2]. With problem comes solutions as we humans strive to fight against our problems. Also, it can be noted that this problem can occur to a small child engrossed most of the times in mobile phones and can range to elders who crave for companionship but no one's around to meet their needs. There have been a number of ways[3] to curb this problem which range from practising physical activities to helping self and others. Many success stories have been written by following such golden steps worldwide. Here, in our project, we propose Man v/s Bot - a multipurpose solution to this global problem that can act as a friend, a companion as well as an entertainer.

**System Architecture:**

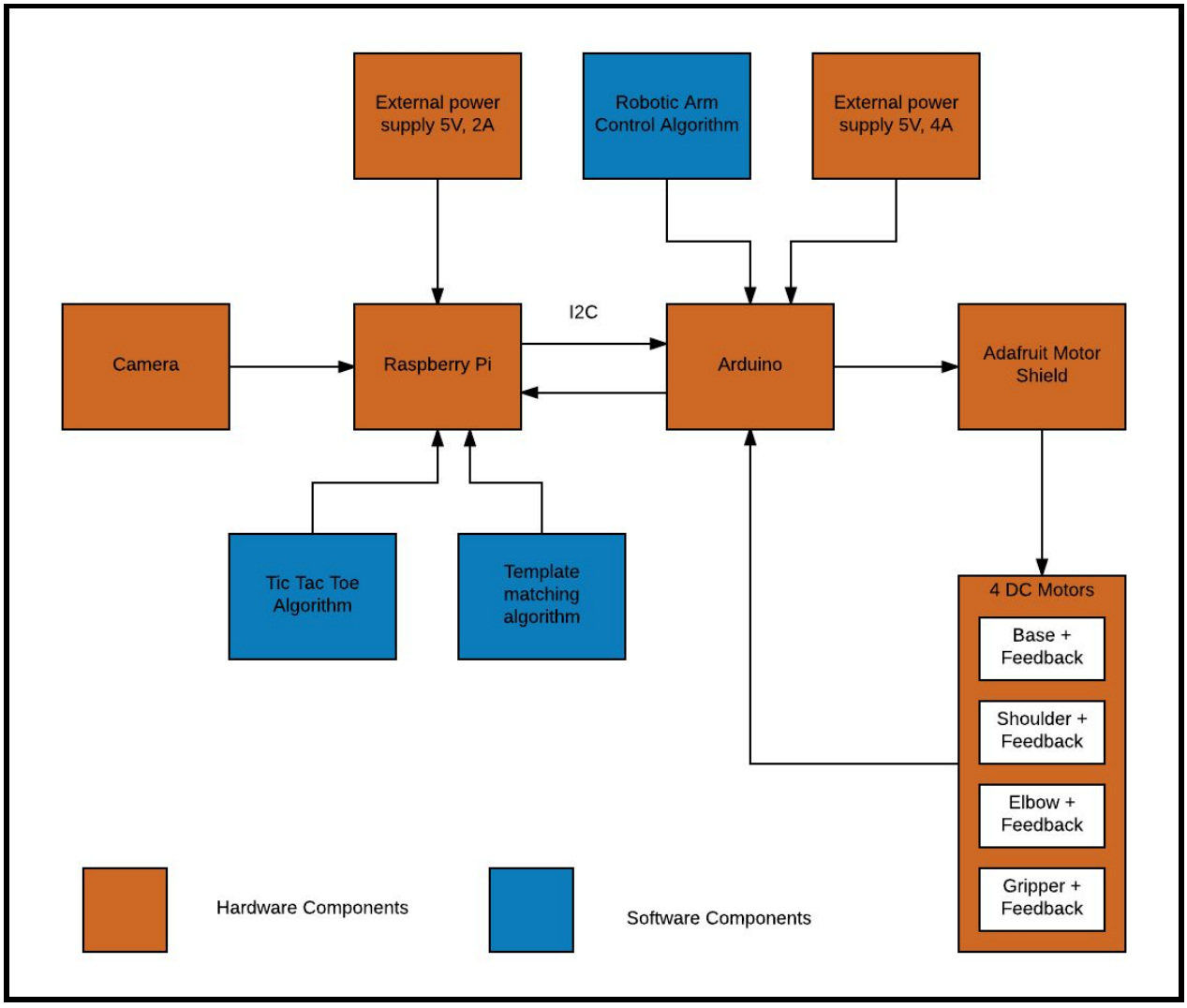



Fig 1: System Block Diagram

Hardware:

The hardware architecture consists of an Arduino microcontroller, that is connected to an Adafruit Motor shield. The motor shield integration is done to facilitate controlled motion of the arm based on the feedback mechanism installed onto each motor (see Fig.2). Thus, the data exchange between the 4 motors on ARM viz. Base, elbow, shoulder and gripper and the arduino helps in traversing to the exact location. A raspberry Pi is used to connect to an external camera. The camera constantly sees the situation on board and based on the template matching software algorithm and the Tic-Tac-Toe artificial intelligence, the raspberry Pi gives a number (position on the grid ranging from 1-9). The RPI communicates with the arduino to give this position via I2C mechanism[5]. This helps the robotic arm to traverse to the set destined location.

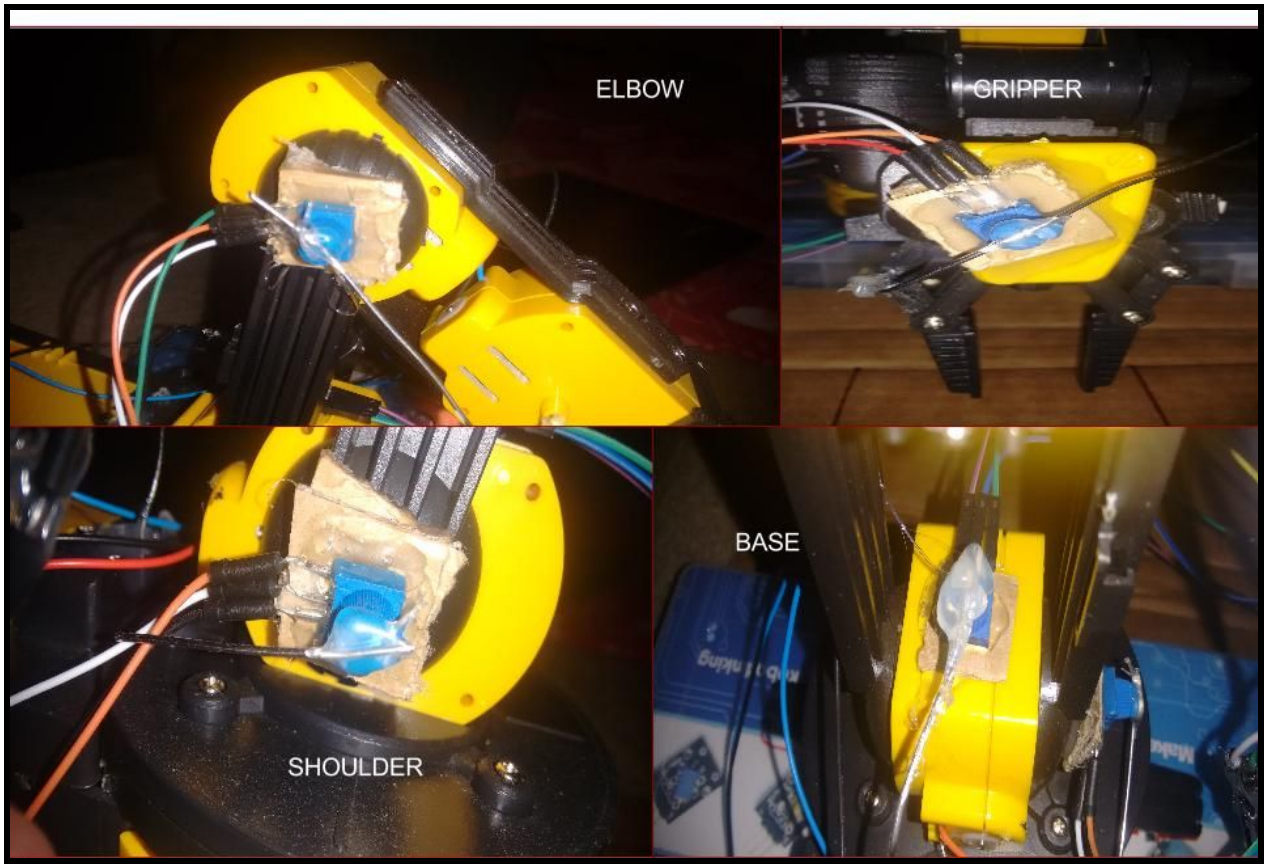


Fig 2. Feedback potentiometers position on each motor

Based on the feedback of various potentiometers, we managed to get a range of values for each motor. Thus, we track these range of values to monitor the movement of arm from base location to the given position in the 3x3 grid of tic-tac-toe.

The below image gives a brief overview of the following:

1. Position mapping on the 3x3 grid.
2. Pin Connections for future reference.
3. Motor connections to the motor shield
4. Base position statistics and range of values of various motors at each position on 3x3 grid (after calibration).

```
/*
_________________
 3  |  6  |  9
____|_____|____
 2  |  5  |  8
____|_____|____
 1  |  4  |  7
____|_____|____
      ARM
_________________

*/

/*
pin connections:
motor M1: orange--> gnd; red-->vcc
motor M3 : orange-->gnd ; yellow-green--> vcc
motor M4: orange-->gnd; black-->vcc
motor M5: purple-->gnd; green-->vcc
*/

//Shield configuration
/*
 * Motor M1 - Gripper:> Backward(IN), Forward(OUT)  yellow-->blue ---driver(M1)
 * Motor M2 - Fixed
 * Motor M3 - Elbow:> Backward(UP), Forward(DOWN)  blue-->green ---driver(M2)
 * Motor M4 - Shoulder:> Backward(DOWN) Forward(UP) orange-->black  ---driver(M3)
 * Motor M5 - Base:> Backward(Left) Forward(Right)  orange-->blue ---driver(M4)
 */
//pickup value: M1-->820    M3-->354    M4-->403    M5-->279
//gripper holding value: M1-->820
//gripper releasing value: M1-->850
//zero position: M3-->325; M4-->511

//mapping for each position:
//pos1:M3-->354    M4-->399    M5-->328(left)-388(right)
//pos2:M3-->511    M4-->271    M5-->369(left)-413(right)
//pos3:M3-->706    M4-->153    M5-->396(left)-454(right)
//pos4:M3-->354    M4-->399    M5-->388(left)-487(right)
//pos5:M3-->511    M4-->271    M5-->454(left)-492(right)
//pos6:M3-->706    M4-->153    M5-->454(left)-477(right)
//pos7:M3-->354    M4-->399    M5-->487(left)-581(right)
//pos8:M3-->511    M4-->271    M5-->492(left)-549(right)
//pos9:M3-->706    M4-->153    M5-->477(left)-525(right)
```

Fig 3. Calibration values and position mapping

**Implementation Flow (Project Milestones):**

| Serial Number | Milestone | Deliverable | Description |
|---|---|---|---|
| 1 | Feedback of arm movement on a console (Serial-Monitor) | Video demonstration of Arm movement and corresponding change in value of potentiometer | We connected potentiometer with a rigid metal rod to one of the motors (see Fig 2). The motor was given supply and the rigid rod caused a change in value of the potentiometer and we noted the corresponding changes displayed on a serial monitor of Arduino. |
| 2 | Adding the motor shield for controlled movement | Video demonstration of moving the motor in a controlled manner using the shield | Here, we connected all the feedback potentiometers to the arm and connected all the motor power supply pins to the Adafruit Arduino motor shield similar to the one explained in milestone 1. We could now control the motion of the motor based on the time for which we supply power to the DC motors. |
| 3 | Calibrating the motor shield for definitive arm movement (in degrees) | Video demonstrating arm movement when given a specific position on the board | Here, we move the motors to the required positions on the tic-tac-toe board by manually giving power supply to the motors and note down the values of all the potentiometers for each position. |

| | | | |
|---|---|---|---|
| | | | After the calibration is done, we put all those values in the code to make the arm movement automated based on the position given as an input. Calibration is done using potentiometers in 4 locations viz. Base, Elbow, Shoulder and Gripper |
| 4 | Robotic Arm movement based on feedback from camera | Video demonstrating the integration of hardware and the software taking feedback from camera | Here, the software team used template matching to get the current state of the board. After the current situation is analysed, the next move of the robot is predicted. The gripper potentiometer is checked if the block is held by the robotic arm or not. If the block is available in the gripper, the arm can move to the position given by the software and place the block at that position. |

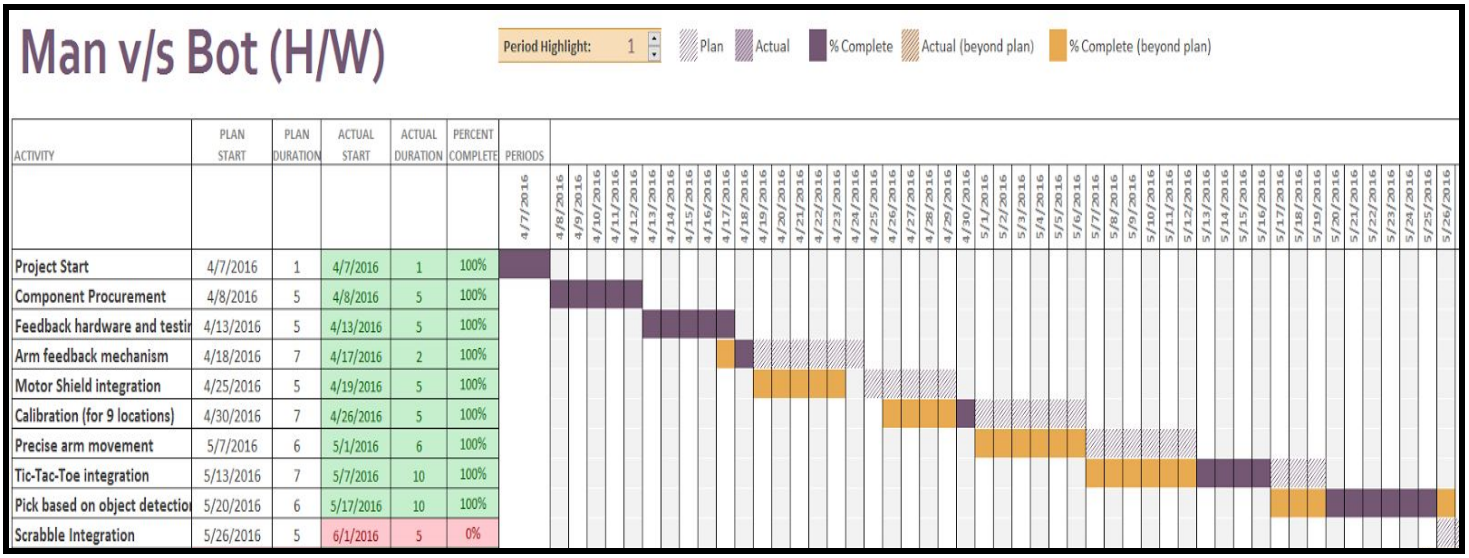
For all the deliverables, please check out our Bitbucket Repository.

**Technical Difficulties :**

- Initially we had powered the Adafruit motor shield using the Arduino Uno Board which was supplied power through the USB port of a laptop. USB 1.0 and 2.0 can deliver a maximum current of 500mA and USB 3.0 can deliver a maximum current of 900mA[4]. The base motor had to drive the maximum load and thus, the current was not enough to power it. To tackle this problem, we used a wall charger with the rating of 5V and a maximum current of 4A.

- Also, there was calibration error for the feedback potentiometers. We checked the value of the feedback potentiometer in a while loop and moved the dc motor in the appropriate direction to make the error zero. We kept the speed of the dc motor constant. We applied a heuristic approach to tackle this issue; the amount of time for which the current is supplied was kept high initially after which we reduced it to a very small value to reduce the overshoot when the robot tries to go to a designated position.
- The Tic-tac-toe implementation had 3x3 grid for which calibrating the robotic arm was possible such that the arm could maneuver over the entire grid. If the board is scaled to a grid of more squares which is the case in board games like chess or scrabble, it can cause a calibration problem as the squares are quite close to each other and thus the granularity of the board increases. The current robotic arm doesn't have the ability to achieve such fine granularity with precision as the movement of base motor is in circular motion and that required for high precision pick and place is linear motion over a grid.
- The arm has a few physical limitations in terms of its workspace envelope. The horizontal and vertical reach of the robot is limited because of the limited angle by which a dc motor can rotate. Also, the potentiometers along with the rigid metal rod connected to the motor to take feedback impose further restrictions of the dc motor movement.

**Conclusion & Future Scope:**




| ACTIVITY | PLAN START | PLAN DURATION | ACTUAL START | ACTUAL DURATION | PERCENT COMPLETE |
|---|---|---|---|---|---|
| Project Start | 4/7/2016 | 1 | 4/7/2016 | 1 | 100% |
| Component Procurement | 4/8/2016 | 5 | 4/8/2016 | 5 | 100% |
| Feedback hardware and testir | 4/13/2016 | 5 | 4/13/2016 | 5 | 100% |
| Arm feedback mechanism | 4/18/2016 | 7 | 4/17/2016 | 2 | 100% |
| Motor Shield integration | 4/25/2016 | 5 | 4/19/2016 | 5 | 100% |
| Calibration (for 9 locations) | 4/30/2016 | 7 | 4/26/2016 | 5 | 100% |
| Precise arm movement | 5/7/2016 | 6 | 5/1/2016 | 6 | 100% |
| Tic-Tac-Toe integration | 5/13/2016 | 7 | 5/7/2016 | 10 | 100% |
| Pick based on object detectio | 5/20/2016 | 6 | 5/17/2016 | 10 | 100% |
| Scrabble Integration | 5/26/2016 | 5 | 6/1/2016 | 5 | 0% |

We could achieve the required granularity and precision for covering all the destinations in a 3x3 grid required by TicTacToe. The same can be achieved with finer granularity for games like chess and scrabble. To facilitate linear motion and avoid a circular motion that induces some error in placing, we can incorporate a maneuvering mechanism such as an iRobot, on top of which the arm can sit. This will help in achieving the required precision.
Custom board game arenas supporting the arm's degree of freedom and workspace envelope can also be developed for better performance.

**References:**

[1]http://www.independent.co.uk/life-style/health-and-families/features/the-loneliness-epidemic-more-connected-than-ever-but-feeling-more-alone-10143206.html
[2]http://www.theregister.co.uk/2006/05/15/loneliness_index/
[3]http://www.huffingtonpost.com/margaret-paul-phd/7-ways-to-avoid-lonelines_b_4999225.html
[4]http://www.extremetech.com/computing/115251-how-usb-charging-works-or-how-to-avoid-blowing-up-your-smartphone
[5]https://oscarliang.com/raspberry-pi-arduino-connected-i2c/