Android phone controlled robot using Bluetooth

Yash Bhalgat | Navjot Singh | Meet Shah
{13D070014   130110071   13D070003}

Group No. T7, Batch - Tuesday
TA : **Sai Sumanth**

Project description:

This project is done using the Pt-51 board, HC-05 Bluetooth module and android phone with accelerometer. User controls the speed and direction of the bot by tilting the phone. Phone transmits the accelerometer values along three axis to the paired Bluetooth module. The HC-05 embedded bluetooth module is configured at a baud rate of 9600 Hz which is set using AT commands in work-response mode.
Further communication is done in auto-connection mode.
HC-05 Bluetooth modules sends the received values to PT-51 through UART. The values obtained through UART communication are processed to extract the x, y, z components of the accelerometer. A logical decision is thus made on the speed and direction of motor turns so that the bot moves in the desired direction, with the desired speed.

Deliverables:

The bot is controlled using two motors (L and R) which derives signals from PT-51 board generated accordingly from the extracted x, y, z values from the android device (via Bluetooth transmission).
The bot can traverse the 2-D surface along 8 independent directions (each separated by an angle of 45°). (Refer to the image below). Furthermore, the speeds in forward and reverse directions can be brought to half of the max. speed using PWM technique. Each direction (region) is also displayed on the LCD interfaced to board.

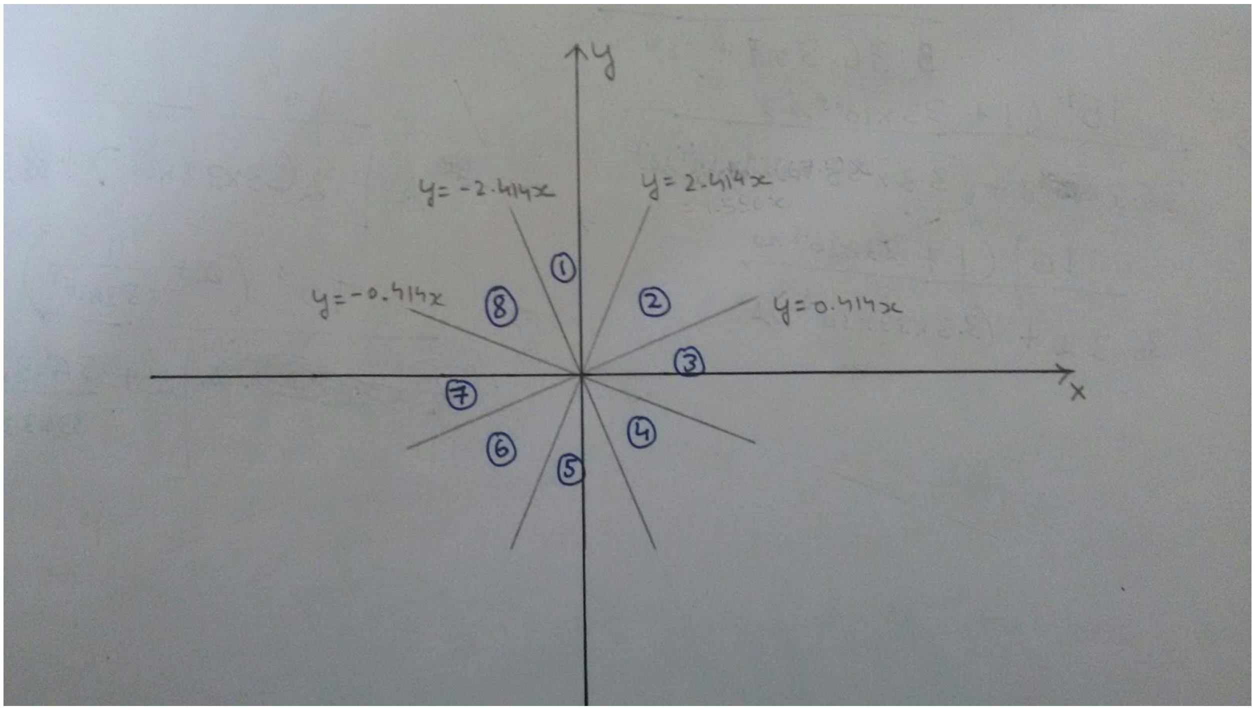


Fig. The surface 2-D plane divided into 8 different regions for direction control.

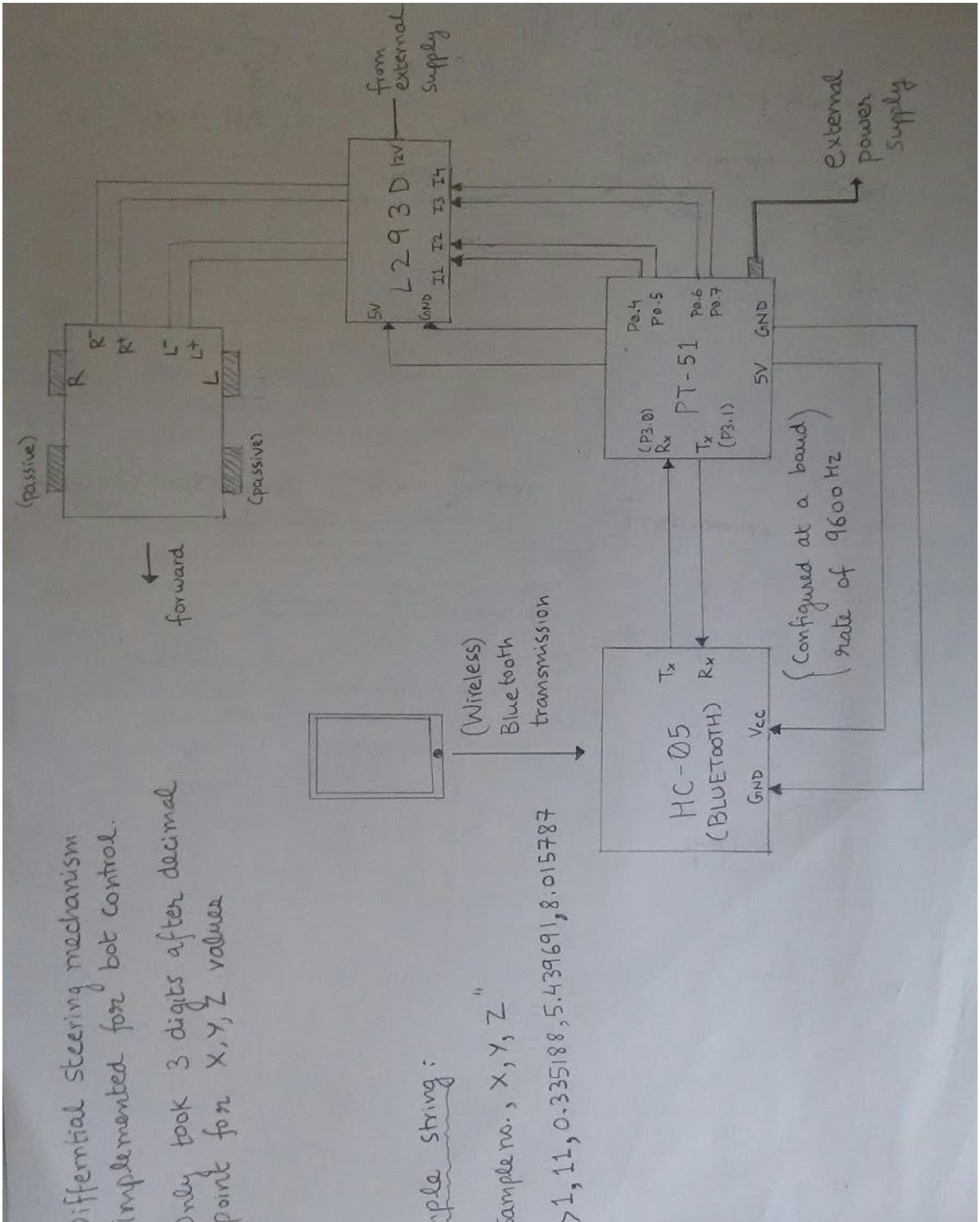
Description of the design:

(Please refer the Block diagram following this discussion for reference.)
The design consists of the following components.(Comments and descriptions are provided for each block) :

(i) Android Device : Used to send the accelerometer values wirelessly to the HC-05 bluetooth module using the app SensoDuino. The values take the form of a string given by :
A sample string consists of : “ >1 , sample number , X , Y , Z “
e.g “ >1 , 12 , 5.012532154 , 4.4124156715 , -1.0312415125 “

(PS : The values are signed. The no. of digits in X , Y , Z may vary !)

Block-diagram:



(ii) HC-05 Bluetooth Module : The device is adjusted (through AT commands) to perform serial communication with the PT-51 board at a Baud Rate of 9600 Hz. This module includes Rx, Tx pins to perform this operation.

(iii) PT-51 board : Used to process the strings (as exemplified above) obtained from serial communication to extract X,Y,Z values. In the code, we have used four 1-bit variables, namely : mot1, mot2, mot3 and mot4. These are connected to pin P0.4, P0.5, P0.6 and P0.7 respectively. The different values these variables take are shown below :

Condition (on x, y, z)	mot1 (L+)	mot2 (L-)	mot3 (R+)	mot4 (R-)	Region (on LCD)
|z|: >9 x: X y: X	0	0	0	0	0
|z|: <7 -y>2|x|	1	0	1	0	1
|z| >7,<9 -y>2|x|	0.5	0	0.5	0	f
|z| <9 -x< -2y -y< -2x	1	0	0	0	2
|z| <9 -x > 2|y|	1	0	0	1	3

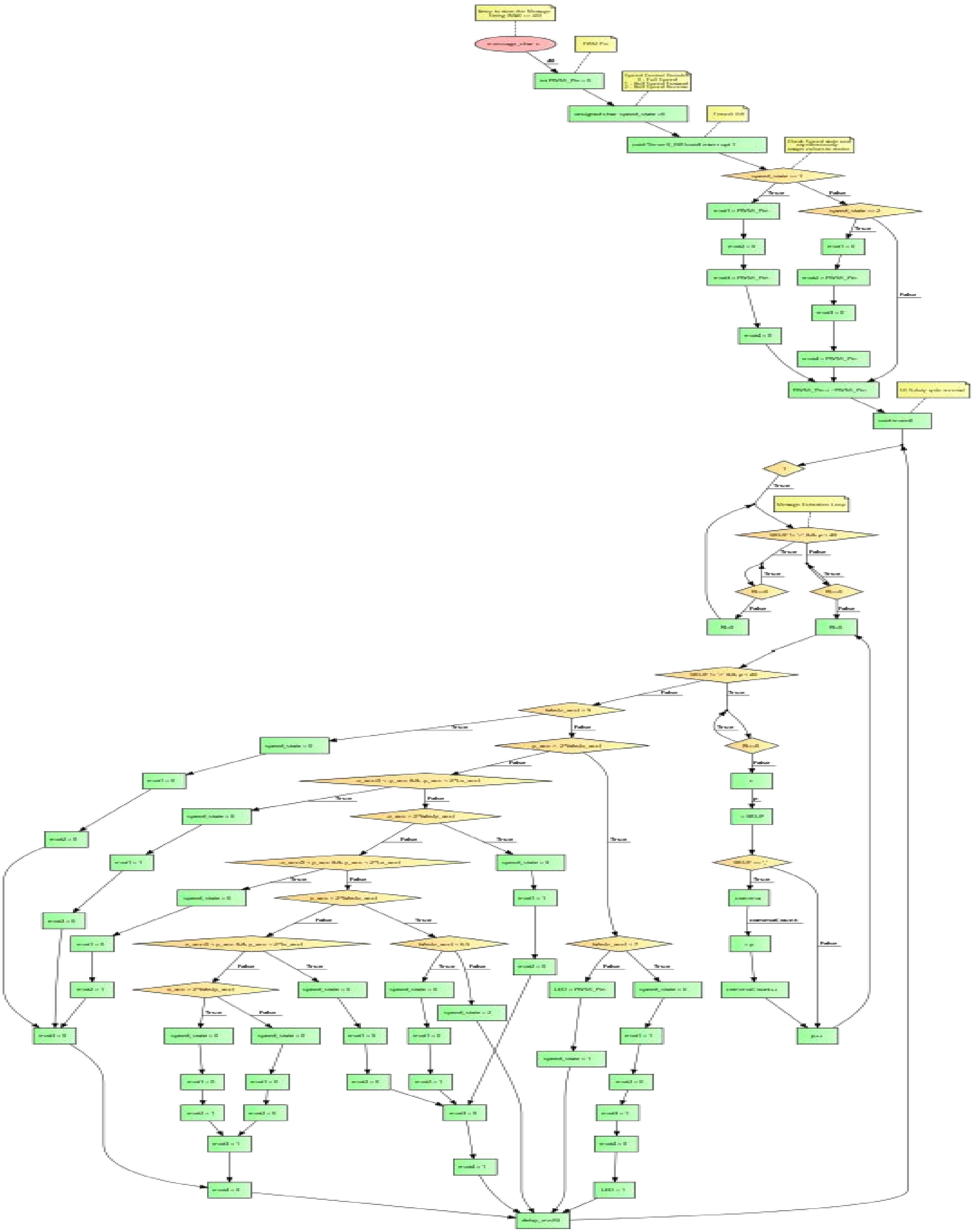
$\lvert z\rvert < 9$ $-x < 2y$ $y < -2x$	0	1	0	0	4
$\lvert z\rvert < 6.5$ $y > 2\lvert x\rvert$	0	1	0	1	5
$\lvert z\rvert: < 9, > 6.5$ $y > 2\lvert x\rvert$	0	0.5	0	0.5	r
$\lvert z\rvert < 9$ $x < 2y$ $y < 2x$	0	0	0	1	6
$\lvert z\rvert < 9$ $x > 2\lvert y\rvert$	0	1	1	0	7
z: < 9 $x < -2y$ $-y < 2x$	0	0	1	0	8

P.S : A state of 0.5 denotes a PWM speed (with 50% duty cycle).
These control signals decide the speed and direction of the motor.

(iv) IC L293D : Used to convert the signals mot1, mot2, mot3 and mot4 to motor specific voltages (mostly 12 V). Requires 5V Vcc and 12V Vref for interpolation. This IC also contains enable pins, be sure to check they are set HIGH.

(v) Motor : We have used only two motors (L and R) to drive the four wheels. Differential steering mechanism is used to turn the bot in different directions and control the speed. There are 4 inputs to the bot, namely, L+, L-, R+ and R-

Flow-chart of the program: (Click here for a High Res SVG image)



Challenges faced and outcome:

One of the few challenges that we faced were as follows :

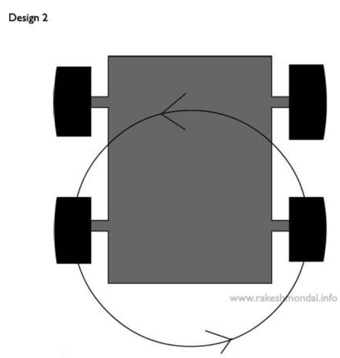
1. **Asynchronization of the PWM Speed select module** :
   As we know from the flowchart the timer interrupts generate the 50% Duty Cycle PWM signal in order to drive the motors at half speed. Since the direction of motion and speed is decided every time a new data string comes from the bluetooth module **HC 05**, the signals to the motor are essentially sent at the rate of the baud rate of the bluetooth module. Now the Frequency of the PWM signal is much much lower than the baud rate and hence if a synchronous assignment of the motor signal values is made, the motor essentially runs at full speed as the signals are determined by the Baud rate code block rather than the timer interrupts . The solution that we implemented was that we made the PWM variable asynchronously assigned and a speed state variable which would be enabled only in those regions (x,y,z) where we need 50% speed and the assignment of the PWM value to the motor signal would now occur in the timer interrupt **TI**.

2. **Extraction of message string from stream data :**
   The extraction of a single message string from the stream data was essential in the region determining logic of the code. One major problem however was to determine the logic to get and recognize the first character of the string and end the string fetching when the next string begins in a live stream data. We ended up traversing to the data feed until a **‘<’** until the next **‘<’** arrived and then extract the required coordinates from the string with string processing.

Shortcomings/ Future improvements:

(i) Limitations due to design :

- No. of motors : The chassis provided consisted of only two motors to drive the four wheels (thus, two of them being passive) . This restricted our vehicle to perform efficient 360 degree turns as we were using differential steering on only two motors.



Improvement : If used a model which supports 4 motors (one for each wheel). We could have assigned the turning roles to the rear wheels (as depicted in the figure). The forward wheels then control the speed and the rear wheels can take on the job of assigning a direction. This provides greater degrees of freedom in movement and provides efficient control.

- Direction Control : We have coded an implementation in which the state of the bot could be from any of the 10 different states (8 direction + 2 PWM) of motor speed. These can be tinkered with to provide more control for turning, steering or speed.(Note that this will work independent of the previously mentioned modification).

(ii) We have made a wired bot, thus,it is limited to operable within a certain fixed range. On board power supply can be used to power the L293D IC and the PT-51 board and we can wirelessly control the car within Bluetooth range.

(iii) PWM mode for each direction can be provided. Here we have done it only for the forward and reverse cases. We may choose more than 2 speeds by implementing other duty cycles.