PROJECT REPORT ON

OBSTACLE DETECTION WITH BLUETOOTH CONTROLLED VEHICLE MOTION

BY

Sasank Das Alladi
Naga Aiswarya Vadlamani
Priyadarsini Pethanaraj
Rohit Chaitanya Kunkumagunta

ECE 511: MICROPROCESSORS

CONTENTS

LIST OF FIGURES

1. ABSTRACT

The purpose of this project is to build a vehicle, controlled through hand gestures with an integrated obstacle detection system. The accelerometer attached to the hand is responsible for the movement of vehicle and will stop if any object is detected, thus avoiding the crash, irrespective of the forward direction. The end picture we intend to achieve will allow the user to control the vehicle more flexibly, when compared to the remote control based system. The obstacle detection system enables crash avoidance within the range of 15 centimeters.

2. MOTIVATION

Toy cars have been the greatest thing to play with for children. Remote controlled cars were first introduced which became popular in brief time. Apart from remote controlling the car, an alternate technology is by hand gestures. Hand gesture technology is an emerging innovation in recent times and has replaced the concept of a remote controller. With the movement of our hand, we can define the action of the car. To this, we can add an extra feature of obstacle detection, i.e. the car stops its movement when it detects any obstacle and prevents it from damaging that is similar to unmanned vehicles. The idea can be developed further to build robots that can be navigated to move things, fetch objects etc.

3. SOLUTION

The set-up uses two MSP430s: MSP430G2, MSP430FR6989. The accelerometer attached to the user's hand decides the direction of the movement of the car. The signals from the accelerometer are transmitted to the MSP430G2, this comprises of the transmitting side. The signal then reaches the MSP430FR6989 through the Bluetooth module HC05. The H-bridge and the Ultrasonic sensor, is interfaced with MSP430FR6989, which now comprises of the receiving side. The DC Motor is connected to the H-bridge which is what drives the car. Ultrasonic sensors are used to detect obstacles within a range of 15cm.

4. BLOCK DIAGRAM

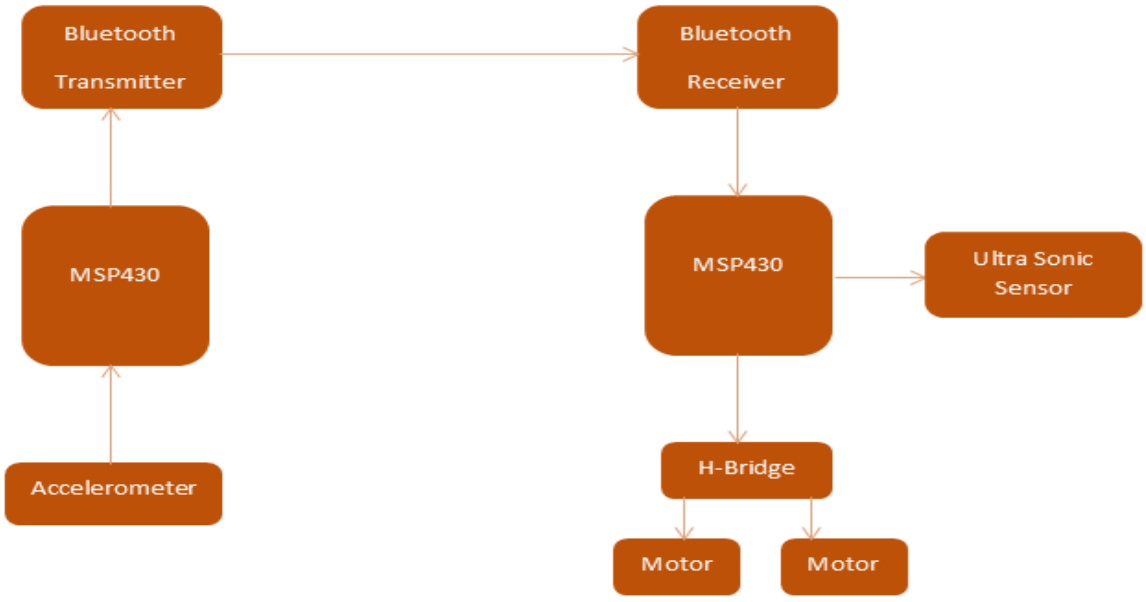

Fig 1: Block Diagram of the Project

5. COMPONENTS DESCRIPTION

5.1 REMOTE SIDE CONNECTIONS

5.1.1 MSP430G2

The MSP430G2 is the main component of the transmitting side. The code is stored and processed in the MSP. The hardware interfacing of the accelerometer is done by connecting X-axis and Y-axis pins to the ports P1_3 and P1_4 (GPIO) of the MSP. The on-board Analog to Digital Converter (ADC) will convert an analog voltage on a pin to a digital number. The slave Bluetooth module is connected to its UART ports P1_1(Rx) and P1_2(Tx).

For software interfacing, we get the analog values from the accelerometer to MSP430G2. The analogRead function is used to convert the analog signal to digital format. The ADC converts the analog voltage values to Digital values between 0 and 1023. We have tested the hand gestures movement and found out the Threshold values (Experimental Values) for the movement of the vehicle in the serial monitor. As per the threshold values that are received, characters are assigned (Refer table). The UART pins sends these characters to the Slave Bluetooth module using the Serial.write function for transmission.

5.1.2 ACCELEROMETER

The accelerometer which we have used is the ADXL335. It has 5 pins out of which two are used for power supply and the ground, and the other 3 pins are used for the X, Y and Z axis respectively. In this project we are making use of the X and Y axis which are interfaced to the MSP430G2 through the ports P1_3 and P1_4. The X-axis is used for forward and backward motion whereas the Y-axis is used for left and right motion. The table below has the threshold values (Experimental Values) for the movement of the car.

Fig 2: Accelerometer ADXL335 chip

X-AXIS	ADC VALUES
CENTRE (F)	470 (421-510)
FORWARD (W)	> 380 && < 420
BACKWARD (S)	> 511 && < 570

Y-AXIS	ADC VALUES
CENTRE (F)	470 (425- 499)
LEFT (A)	> 380 && < 425
RIGHT (D)	> 500 && < 560

5.1.3 BLUETOOTH MODULE

HC-05 Bluetooth module is used in our project which is a digital component which receives and transmits serial data. For configuring Bluetooth modules, initially both the modules will be in Slave mode. We must configure one module as Slave and the other as Master. By enabling one slave module, we need to get the address of it by using the command AT+ADDR? Then we need to enable another module and change it to Master mode by changing the role of it from 0 to 1 by using the command AT+ROLE=1. Finally, to bind the slave and master we need to bind the slave address to the master module by using the command AT+ADDR = "slave address". By this the wireless communication is successfully configured.

Slave Module:

Hardware interfacing Slave Bluetooth Module, the Rx and Tx of Bluetooth modules are connected to MSP430G2 using UART ports P1_1 and P1_2. Software interfacing Slave Bluetooth Module, we use the Serial.write function to send the respective characters (W, A, S, D, F) to the Slave module.

Master Module:

Hardware interfacing Master Bluetooth Module, the Rx and Tx of Bluetooth modules are connected to MSP430FR6989 using UART ports P4_2 and P4_3. For Software Interfacing Master Bluetooth Module, we use the Serial1.Read function to receive the respective characters (W, A, S, D, F) from the Slave module.

The AT commands used are:

- AT+ROLE – 0(slave), 1(master)
- AT+CMODE- 0 (connect to one) 1(connect to any)
- AT+ADDR- to obtain the slave module's address
- AT+BIND to bind the obtained slave address to the master module

5.2 VEHICLE CONNECTIONS

5.2.1 MSP430FR6989

MSP430FR6989 is the heart of the vehicle. We used the following features of MSP430: UART0 for LCD, UART1 for serial communication (Bluetooth modules to MSP430FR6989), LED1 and 2 for indicating directions. We use the GPIO pins for connecting the H-Bridge and Ultrasonic Sensor.

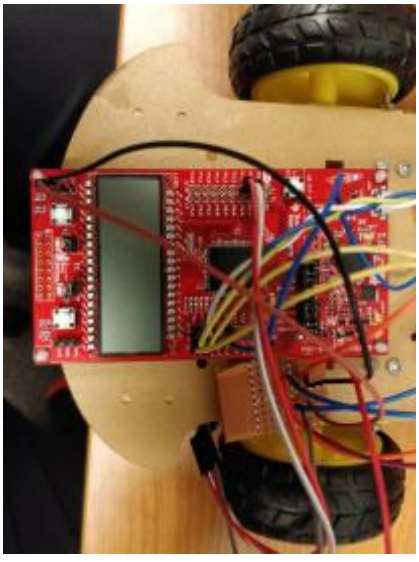

Fig 3: MSP430FR6989 Board layout

5.2.2 H-BRIDGE and DC MOTORS

H-Bridge L293D drives the two D-C motors m1 and m2. The input pins of the H-Bridge are connected to the MSP and the Output pins are connected to the motors 1 and 2. The MSP sends signals through the input pins regarding the motion of the car and the motors connected through the output pins translate this motion.
Hardware Interfacing: Supply and Ground pins from the MSP430 are used to power the H-Bridge. The enable pin connected to pin P2_3 of the MSP is kept high to enable the H-Bridge.
For Motor1 signal: m1pin1 – P3_0, m1pin2 – P3_1,
For Motor2 signal: m2pin1 – P2_4, m2pin2 – P4_7.
For Motor1 drive: H-Bridge pin 3 and pin 6.
For Motor2 drive: H-Bridge pin 11 and pin 14.
For Software interfacing, we use the "pinMode" function to specify each pin as input/output for the MSP430. We use the function "digitalWrite", to set the pins High/Low depending on the logic of the program.

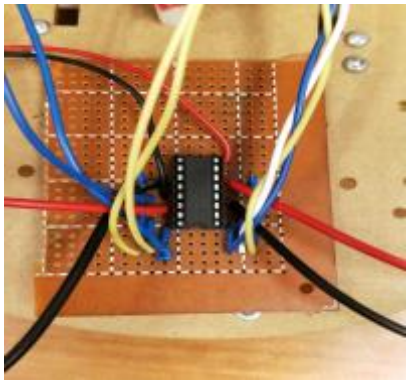

Fig 4: H-Bridge

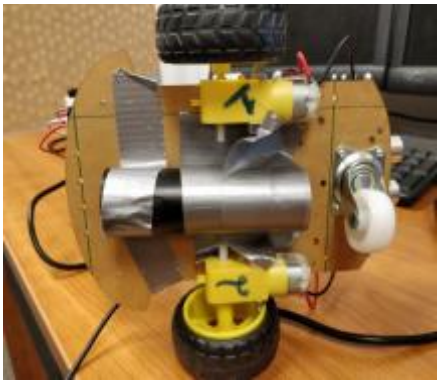

Fig 5: DC Motors

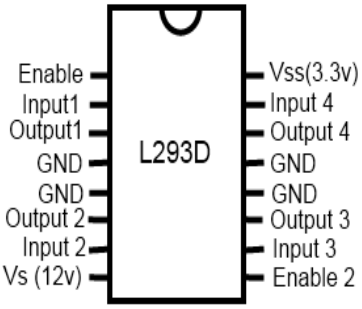

Fig 6: H-Bridge Pinout

5.2.3 ULTRASONIC SENSOR

The ultrasonic sensor used in this project is the HC-SR04. For Hardware interfacing, the sensor's trigger pin is connected to the pin P2_1 and the echo pin is connected to the pin P2_2 of the MSP. For Software interfacing, we use the function "pinMode" to define if the pin is Input/output, here the Trigger pin is in output mode and Echo pin is in input mode. We use the "digitalWrite" function to write the trigger pin to high or low and function "pulseIn" to read the reflected trigger signal off the object. The function "delayMicroseconds" is used to time the trigger signal.

Working: The MSP430 writes the trigger pin for every iteration of its main loop using the digitalWrite function. The trigger signal is sent for 5 microseconds. Then the echo pin is made high to detect the reflected trigger signals (which is the echo). The echo signal which is the time for which the signal was high is read using the "pulseIn" function (this function multiplies the echo time and the speed of sound) and then stores the result in a variable. This value is then passed to another user defined function to convert it to centimeters. The formula used is,

Distance = microseconds / 29 / 2

Then according to the distance value calculated the MSP has been programmed to work as specified in the code i.e. if the object distance is less than 15cm, then the vehicle does not move forward.

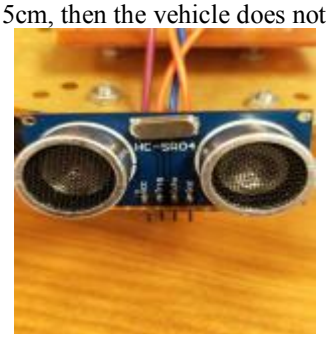

Fig 7: Ultrasonic sensor HC-SR04 Pin layout

5.2.4 FINAL VEHICLE CONNECTIONS

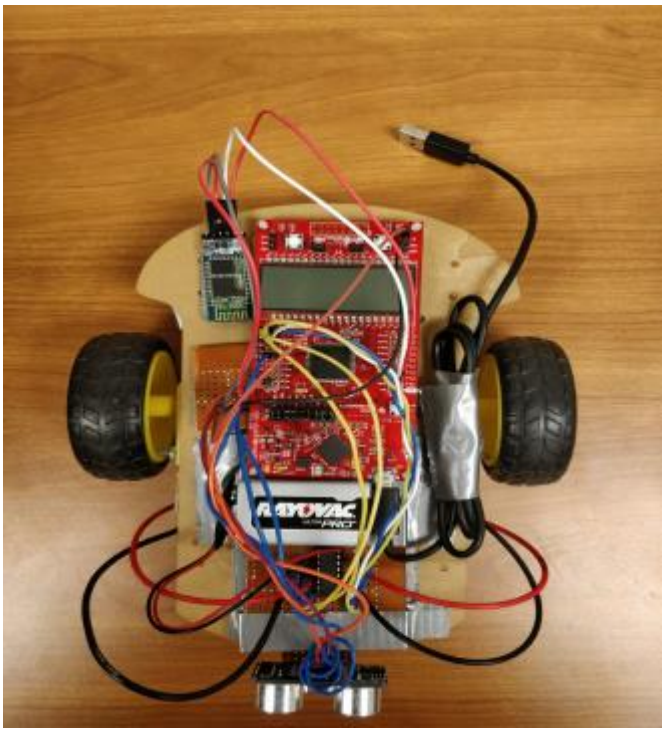

Fig 8: Vehicle side Connections

6. FINAL OVERVIEW

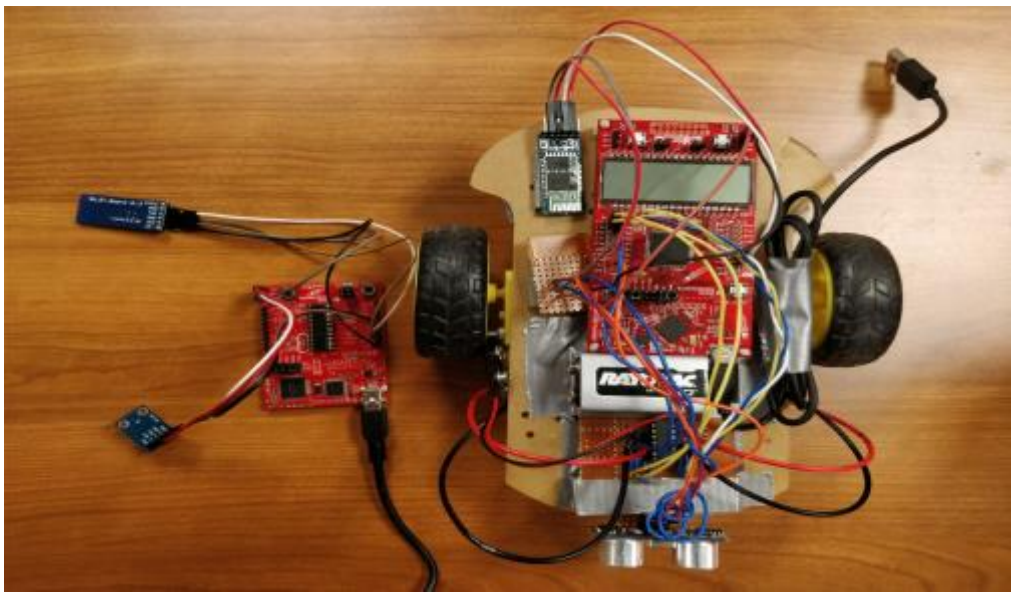

Fig 9: Final Overall view of the Project

7. RESULTS & CONCLUSIONS

The components were purchased, and individual component testing was done to ensure the functionality. The analog values were converted to digital format using MSP430 (ADC) and it was wirelessly transmitted through the Bluetooth module to the MSP430 in the vehicle side connection. H-Bridge and DC motors were interfaced with the MSP430 to make sure that the vehicle moves in the direction of the gesture made. For crash avoidance, we have interfaced ultrasonic sensor to stop the vehicle if the obstacle is within 15cm irrespective of the forward gesture given by the user. As per our project proposal, we have implemented the idea proposed by interfacing all the components successfully.

8. LESSONS LEARNT

We learnt that individual component testing needs to be done before interfacing to make sure that the component is not faulty. The importance to have spare components was also realized. Obtaining the threshold values (Experimental values) from the accelerometer was challenging. We learnt how to use AT commands to interface the slave Bluetooth module with the master terminal module which needed lots of research.

9. ACHIEVEMENTS

The idea we proposed has been successfully implemented without any modifications. The work was divided and it was done as per the schedule. We hope our project can be developed further to be implemented in various industries.

10. BIBLIOGRAPHY

- MSP430G2, MSP430FR6980 Datasheets
- ADXL335 Datasheet
- L293D Datasheet - http://www.ti.com/lit/ds/symlink/l293.pdf
- HC-SR04 - http://www.micropik.com/PDF/HCSR04.pdf
- Energia for coding
- Hype! terminal for serial port communications and AT commands
- Bluetooth module HC-05 https://www.itead.cc/wiki/Serial_Port_Bluetooth_Module_(Master/Slave)_:_HC-05

11. APPENDIX

11.1 DIVISION OF WORK

Sasank Das Alladi: MSP and Accelerometer interfacing.

Priyadarsini Pethanaraj: Bluetooth Interfacing

Naga Aiswarya Vadlamani: Ultrasonic sensor interfacing

Rohit Chaitanya Kunkumagunta: H-Bridge and DC motors.

11.2 LIST OF COMPONENTS

Component Name	Model	Quantity
MSP430	G2, FR6989	1 each
Bluetooth Module	HCO5	2
Accelerometer	ADXL335	1
H bridge	L293D	1
Ultrasonic Sensor	HC-SR04	1
Dc Motors		2

11.3 SCHEMATICS

11.3.1 Remote Side

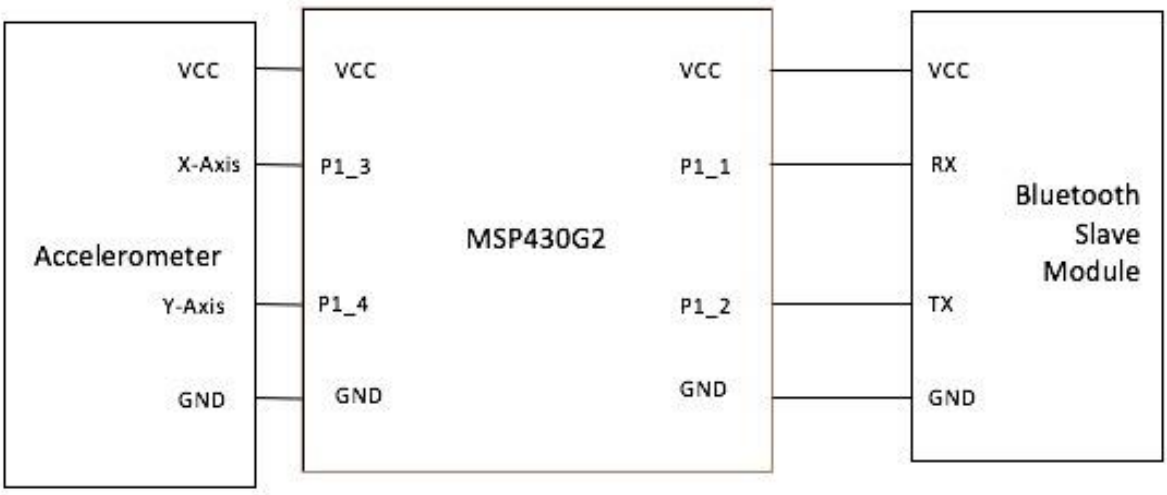

Fig 10: Schematic of the remote side connection

11.3.2 Vehicle Side

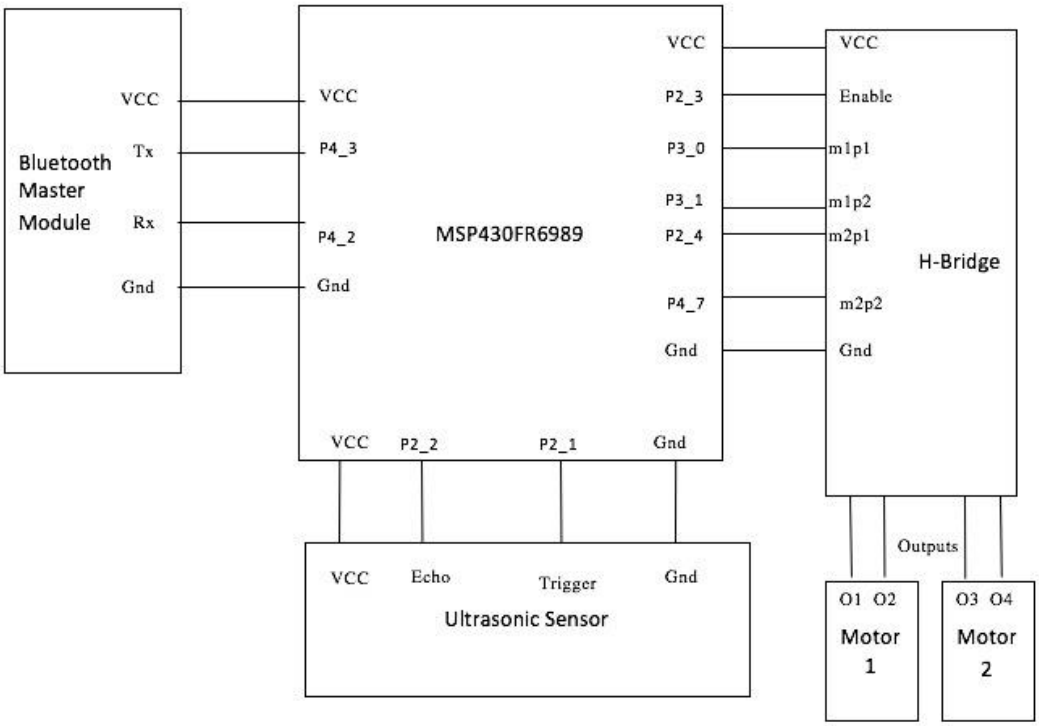

Fig 11: Schematic of the Receiver Side Connection