ECE/CS 6780
Embedded System Design

Project Report
Hand Gesture Robot Control

30th April 2019

Sachin Boban
Bhawna Lakshmipathy
Mohit Kumar
Farhan Nasrullah

1 Purpose

The purpose of this project is to implement the fundamentals of Embedded Systems learned as part of ECE/CS 6780 course.

2 Introduction

Robots are electro-mechanical systems that can be operated by a computer program. Robots can be autonomous or semi-autonomous. An autonomous robot is not controlled by human and acts on its own decision by sensing its environment. A semi-autonomous robot requires some level of human decision making.

Robots requiring high level of precision and speed, are mostly autonomous. But there are applications that are semi-autonomous i.e., some human intervention in one way or the other.

A gesture controlled robot is nothing but a simple vehicle, whose movement is controlled by the gestures or movement of a simple controller. Depending on the gesture made using the controller, the robot can move forward, backward, left, right or halt. The robot can also move forward or backward while making a left or right turn.

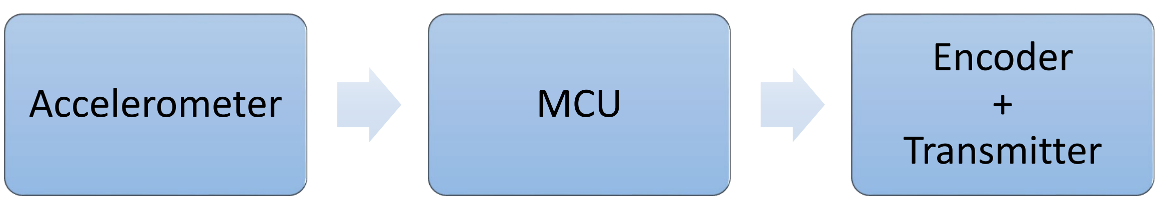
3 Function Block Diagrams

On a very high level, the robot consists of 2 components:

1. Gesture Controller, which is used to control the movement of the vehicle; and
2. Vehicle Controller, which controls the vehicle according to the signals received from Gesture Controller.

3.1 Gesture Controller

The functional block diagram of the controller is as given below:

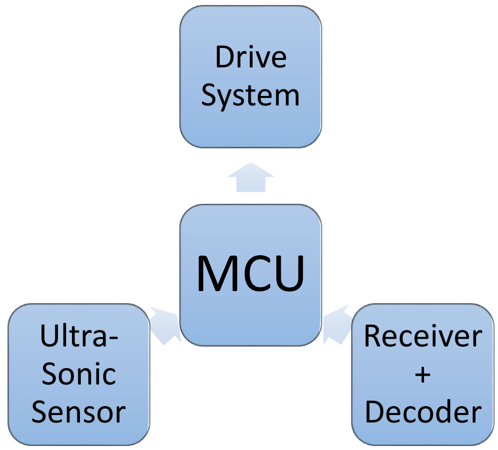


Controller functional diagram

Accelerometer gives the X and Y coordinates of the controller. Based on these coordinate values, the MCU determines the movement of the vehicle. The movement information is then encoded in a specific format and transmitted to the vehicle using an RF transmitter.

3.2 Vehicle Controller

The functional block diagram of the vehicle is as follows:



Vehicle functional diagram

The RF signals once received from the transmitter are decoded. Once decoded, the MCU interprets the movement information and controls the vehicle through the drive system. The ultra-sonic sensor is used to determine presence of any obstacles in the front of the vehicle.

The movement information is conveyed from the controller to the vehicle via RF signals.

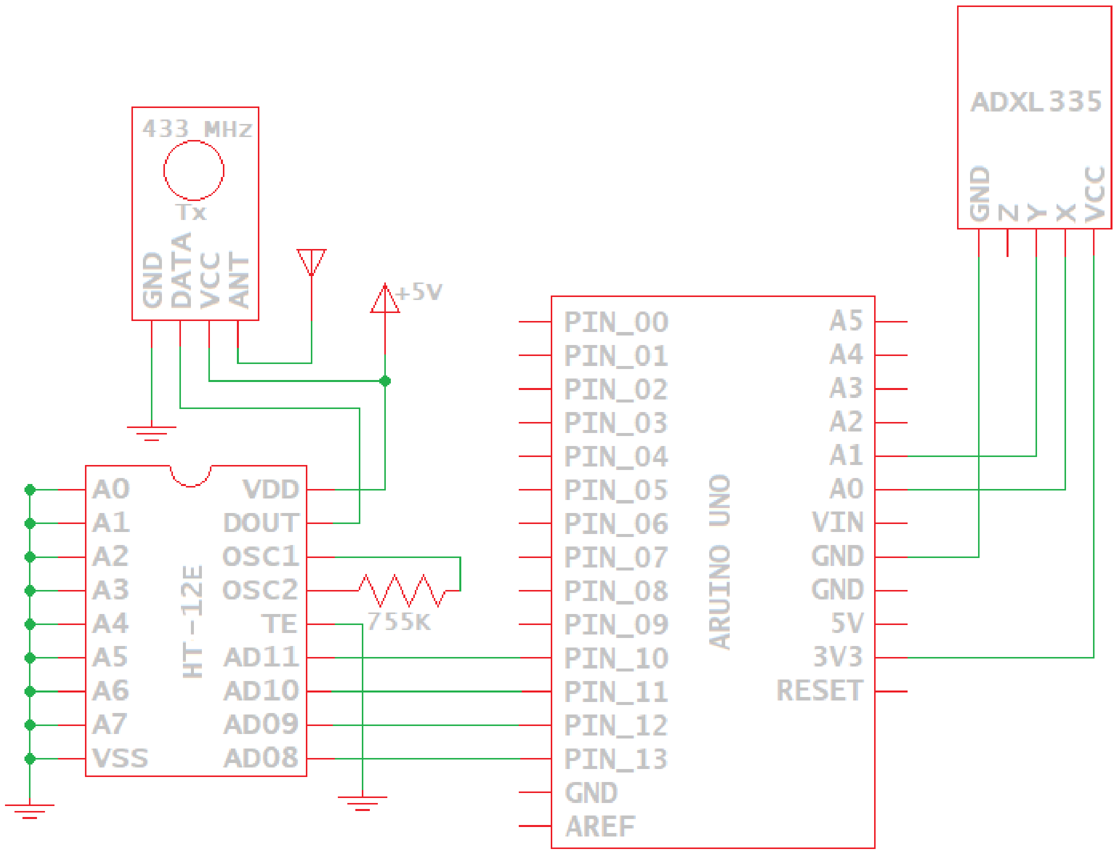
4 Hardware Description

4.1 Gesture Controller

4.1.1 Components

1. Accelerometer: ADXL335
2. MCU: Arduino Uno
3. Encoder: HT-12E
4. Transmitter: 433MHz RF transmitter

4.1.2 Wiring Diagram



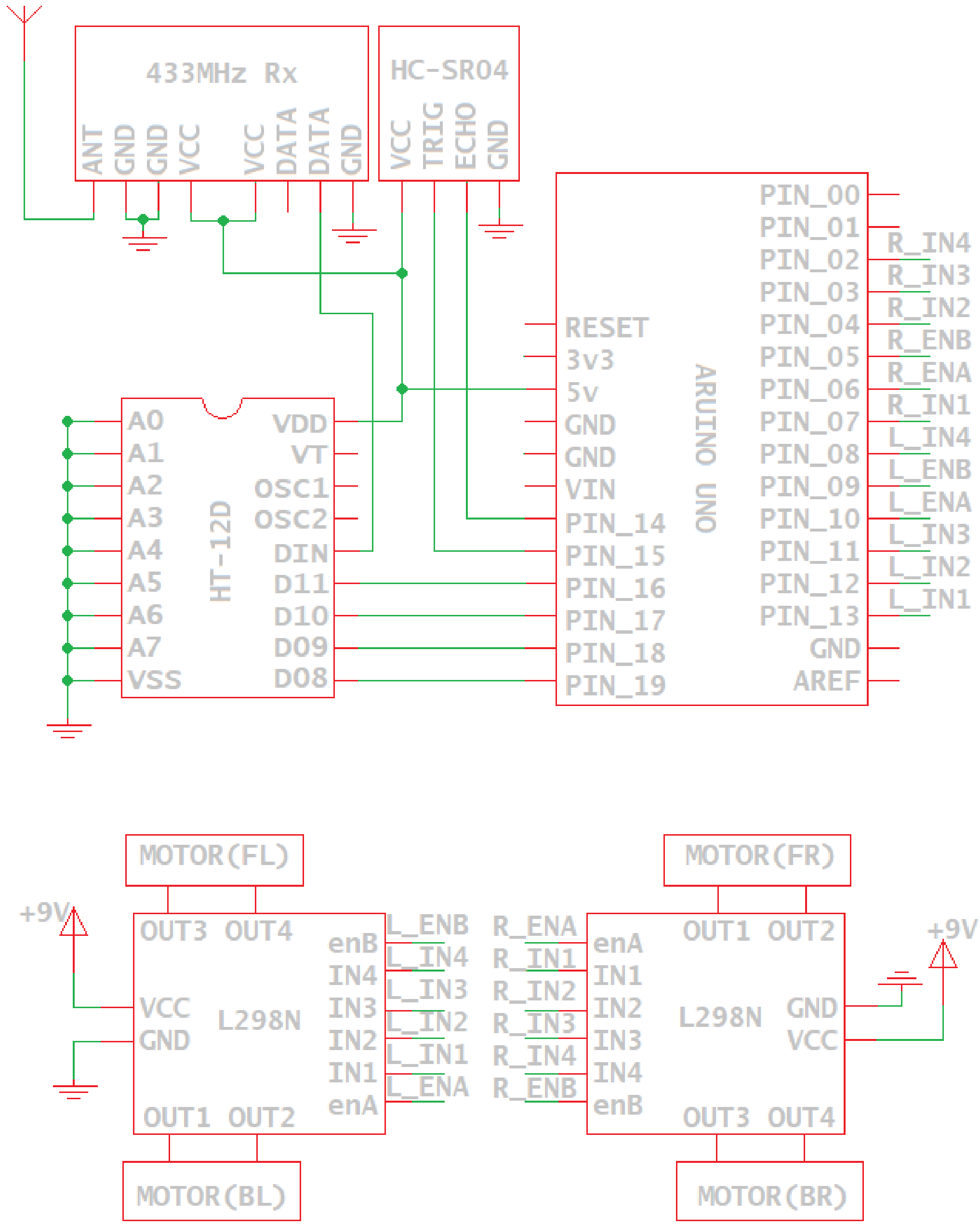
Gesture Controller Wiring Diagram

4.2 Vehicle Controller

4.2.1 Components

1. Receiver: 433 MHz RF Receiver
2. Decoder: HT-12D
3. Ultra-Sonic Sensor (HC-SR04)
4. MCU: Arduino Uno
5. 2 Dual H-Bridge (L298N)
6. 4 DC motors (3-12V)

4.2.2 Wiring Diagram



Vehicle Controller wiring diagram

5 Software Overview

5.1 Gesture Controller

The X and Y coordinates are read from the Accelerometer (ADXL335) through ADC interface. Depending on the coordinate values we determine the movement information to the transmitted.

Directions of movement is determined as the deviation of the X and Y coordinates from their reference coordinates:

- X-axis
  - +ve deviation: Turn right
  - -ve deviation: Turn left
- Y-axis (Forward/Backward)
  - +ve deviation: Move forward
  - -ve deviation: Move backward

A deviation on both the axes conveys a combination of left/right turning along with moving forward/backward. This forward/backward motion while turning left/right is called as sub-direction. The absolute value of the deviation along the Y-axis determines the gear/speed of the vehicle in the forward/backward movement.

Once the direction, speed and sub-directions are determined, we configure the encoder pins as given below.

Similarly, the Encoder (HT-12E) takes 4 pins input. The 4 pins of the Encoder are configured as follows:

- [Enc1 : Enc0]: Direction

Enc1	**Enc0**	**Direction**
0	0	Forward
0	1	Backward
1	0	Left
1	1	Right

- [Enc3 : Enc2]: These pins are configured depending on the direction
  - If direction is forward/backward, these pins represent the gear/speed with which the vehicle shall move forward/backward respectively
  - If direction is left/right, these pins denotes sub-direction (i.e., move forward/backward while turning):
    - [Enc2]
      - 0: No sub-direction, i.e., pure left/right turn
      - 1: [Enc3] shall be interpreted as sub-direction
    - [Enc3]: Sub-direction (valid only if Enc2 = 1)
      - 0: Forward
      - 1: Backward

5.2 Vehicle Controller

The RF signals are received on the RF433 Receiver. These signals are decoded by the decoder (HT-12D). MCU reads the output from the decoder, and interprets the pins to get the direction, gear and sub-direction as mentioned before. The vehicle also detects any obstruction in the forward direction using the Ultra-Sonic Sensor (HC-SR04). If this distance is less than a safe distance, the vehicle is prevented from going forward. Depending on the direction, gear and sub-direction, the drive system moves the vehicle forward/backward, turns left/right, or even turn left/right while moving forward/backward (sub-direction).

Depending on direction, gear and sub-direction information, the Drive system drives the speed and direction of each of the 4 motors (using L298N) as given below:

Movement info		**Motor**							
Direction	Sub-direction	Front-Left		Front-Right		Back-Left		Back-Right	
		Direction	Gear	Direction	Gear	Direction	Gear	Direction	Gear
Forward	-	CLK	gear	CLK	gear	CLK	gear	CLK	gear
Backward	-	ACLK	gear	ACLK	gear	ACLK	gear	ACLK	gear
Left^	-	ACLK	GEAR_1	CLK	GEAR_1	ACLK	GEAR_1	CLK	GEAR_1
Left	Forward	CLK	GEAR_1	CLK	GEAR_2	CLK	GEAR_1	CLK	GEAR_2
Left	Backward	ACLK	GEAR_1	ACLK	GEAR_2	ACLK	GEAR_1	ACLK	GEAR_2
Right^	-	CLK	GEAR_1	ACLK	GEAR_1	CLK	GEAR_1	ACLK	GEAR_1
Right	Forward	CLK	GEAR_2	CLK	GEAR_1	CLK	GEAR_2	CLK	GEAR_1
Right	Backward	ACLK	GEAR_2	ACLK	GEAR_1	ACLK	GEAR_2	ACLK	GEAR_1

*gear: gear as received from the controller.
CLK: Clockwise
ACLK: Anti-Clockwise
^ Denotes a pure left/right turn.

The drive system supports 4 gears:

Gear	**Speed**
GEAR_0	0
GEAR_1	150
GEAR_2	200
GEAR_3	255

6 Milestones

1

- **April 5-8**
- Ordering of parts and familiarisation

2

- **April 9-16**
- Low-level C code (GPIO, ADC) and basic drive system
- Soldering and inter-component connections

3

- **April 17-21**
- Setting up RF communition
- Low-level C code (PWM)

4

- **April 22-23**
- Final connections
- System integration & testing

7 References

1. Arduino Uno
2. ATMEGA328 Data Sheet (MCU)
3. L298N Data Sheet (Dual Full-Bridge Driver)
4. Encoder (HT-12E) Data Sheet
5. Decoder (HT-12D) Data Sheet
6. Transmitter Data Sheet
7. Receiver Data Sheet
8. Accelerometer (ADXL335) Data Sheet
9. Ultra-Sonic Sensor (HC-SR04) Data Sheet

8 Google Slides Presentation

The google slides presentation can be found in GitHub here.

9 Source Code

Code for the project and detailed README can be found in GitHub.

10 Demo

A demo video of the robot can be found here.