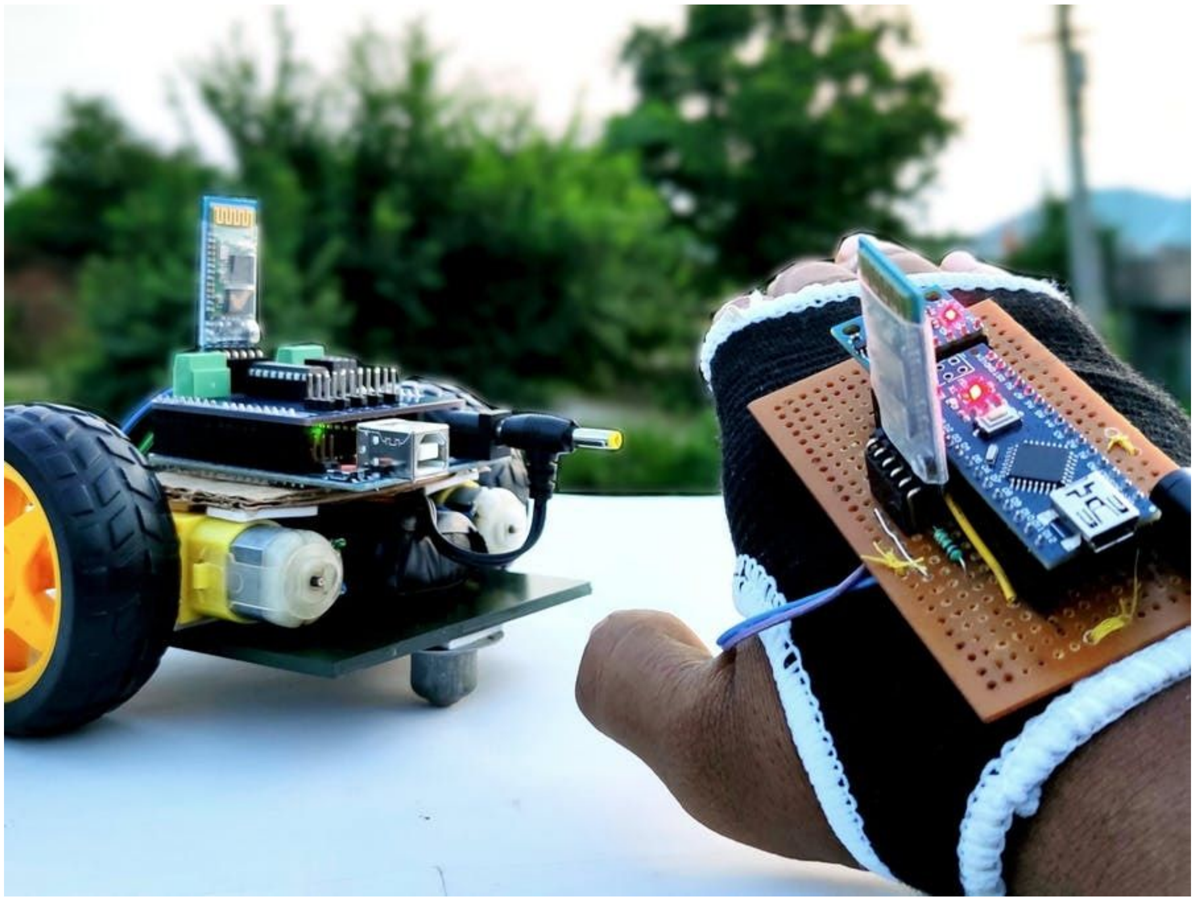
Gesture Control and automatic Parking Car



Mithilesh Thakkar - AU1841038
Yagnik Hingrajiya - AU1841094
Prince Dalsaniya - AU1841124
Hardik Modhavadia - AU1841091

Motivation :

When we see a person who with a physical disability and their wish to taste the thrill and fun of driving a car by themselves, we fell there should be something that can help themselves control without extensive use of hand and leg, and there our product will come into the picture, you would be able to drive the car with just hand gestures and the car will also able to identify the places for parking your car, as many people find difficulty in parking the car from inside, with this technology you will be able to park your car from outside, that makes your car-driving and parking experience more and more fun and easy task rather than a boring one and that drives us to build this amazing car.

Description :

As we know, the gesture is an action performed by a part of the body by someone and he tries to convey a message by that movement. So, we will try to recognize that gesture and control the car. We will achieve this by giving the user a transmitting device. Through which we can record the movement and make a decision. The device will have a sensor called the accelerometer, through the readings of that accelerometer we will set proper levels of reading to encode the information in which direction the sensor moved and transmit that by RF transmitter. On the other hand, we will receive that by RF receiver and decode that by decoder IC. By using that data microcontroller will process and pass instructions to motors to work in some specific manner. Meanwhile, we are recording and calculating the free space for parking on the right side of the car using the ultrasonic sensor. Whenever the gap will be larger than the parking space, we will display that "Parking Space Detected" on the right side. This is the Description of our project.

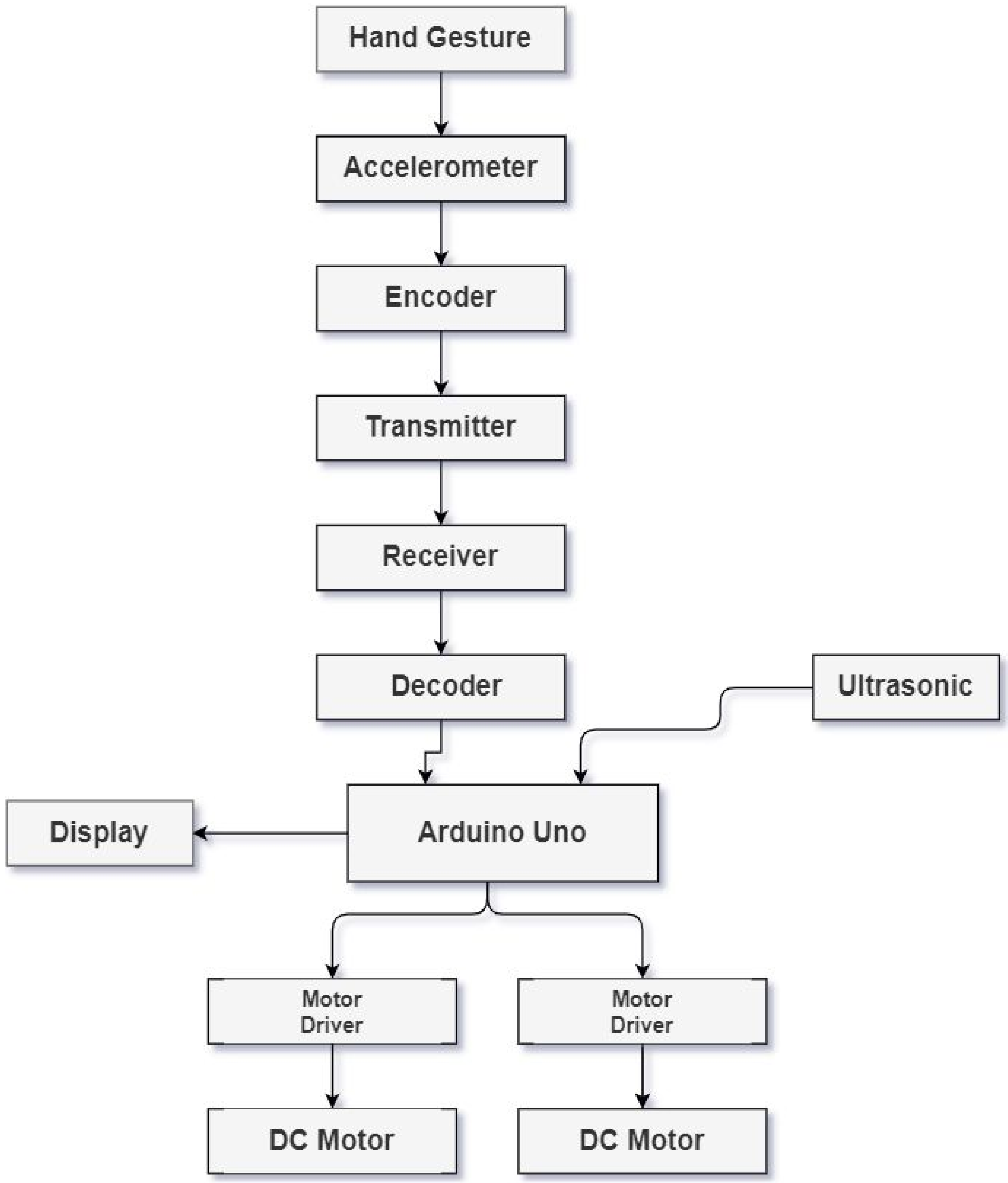
Final Outcome:

As a final outcome, we’ll get a 3 wheeler car that is controlled by hand gestures which can be introduced to you in the form of a glove. And it checks for the availability of empty parking space for you. as space is found it will pop up a message in its tiny display “space available for parking”.

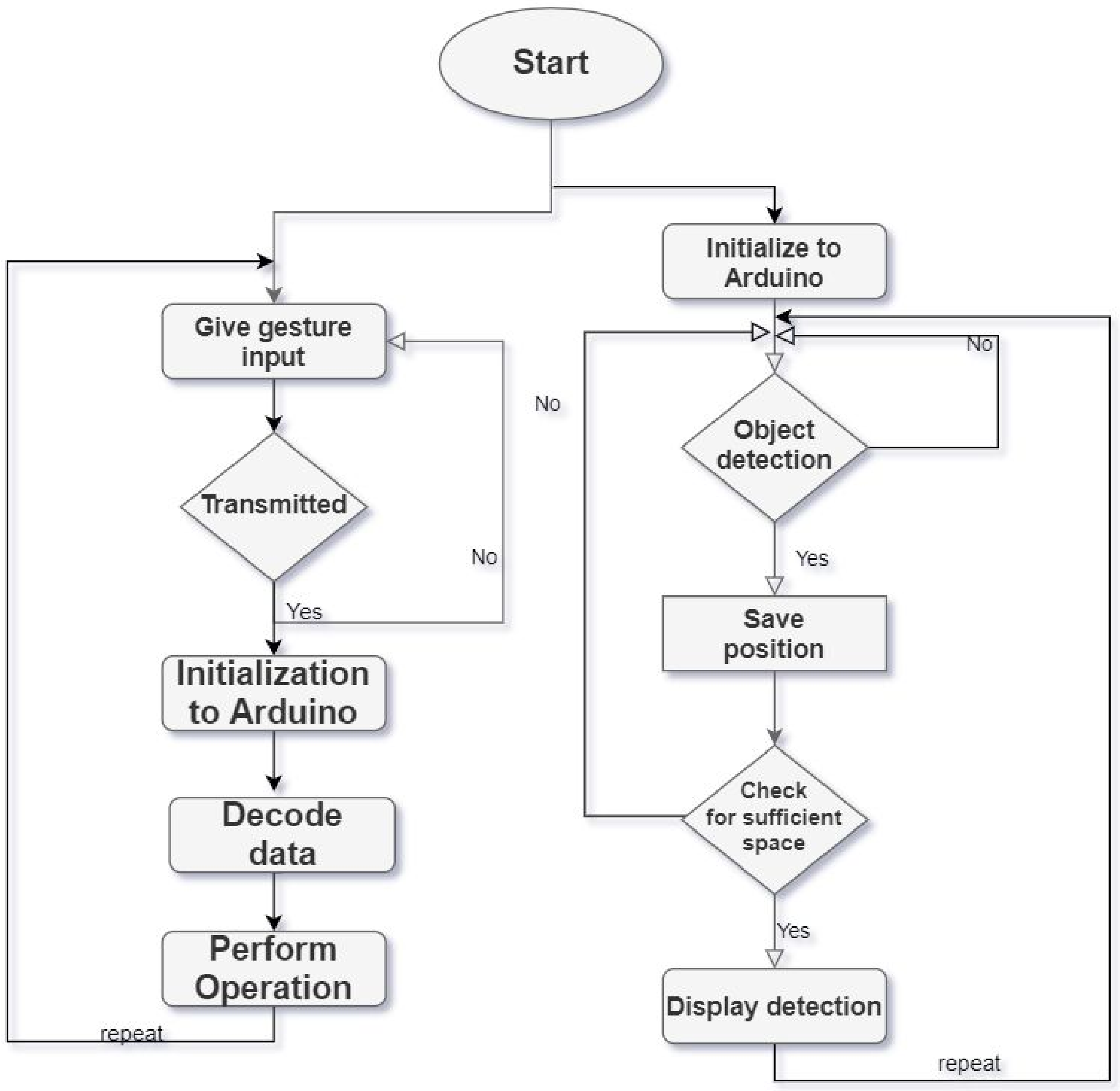
Components

Accelerometer(ADXL335)	1
Arduino Uno	1
HT12E (Encoder IC)	2
HT12D (Decoder IC)	2
RF Transmitter-Receiver	2
L-293D (Motor-Driver)	2
DC Motors (1000 rpm)	2
Display (Character LCD(GMD1621C))	1
H-Bridge	1
Chassis for robot	1
Rear Wheels	2
Front Wheel	1
Jumper Wires	40
Ultrasonic Sensor	2
Power Supply (Batteries 9v)	5
Breadboard	2

Block Diagram



FlowChart


Start
Initialize to Arduino
Give gesture input
No
Object detection
No
Transmitted
No
Yes
Yes
Save position
Initialization to Arduino
Decode data
Check for sufficient space
Yes
Perform Operation
Display detection
repeat
repeat

TIMELINE:

13/2 Submission of Report 1.

5/3 Submission of Report 2.

13/3 Assemble Remote Control.

19/3 Assemble the car and connect the remote control.

26/3 Submission of Report 3

31/3 Finish assembling and start testing and debugging.

10/4 Ready for demo.

15/4 Ready with Final report, poster, car.

SUCCESS!

Selection criteria of components

1. **Accelerometer (ADXL335):-** It is the heart of our Robo which provides us with the position of x,y and z axis of the hand moment.

2. **Arduino Uno:-** It is like the brain of our Robo, so we will use it to dump the code of the working device.

3. **HT12E (Encoder IC):-** To convert the parallel inputs to serial outputs, it will convert our 12 bit parallel data into serial through RF transmitter.

4. **HT12D (Decoder IC):-** We can transmit 12 bit parallel data serially, basically to decode the data received by the receiver.

5. **434 MHz RF-Transmitter:-** It enables us to transmit the data of x,y,z location from the accelerometer to the Robo.
.
6. **434MHZ RF-receiver:-** This is like the ears of out Robo, so we will use it to receive the data from the Accelerometer.

7. **L293D (Motor Driver)**:- As we need one circuit which decides which motor two operate this motor driver will decide which DC motor to start.

8. **DC Motors:-** It will enable the device to move physically when a person will give a command through hand gestures.

9. **LCD Display (GMD1621C):-** To show if the parking spot is available or not.

10. **H bridge:-** we want a catalyst type circuit which enables the DC motor to move forward or backward.

11. **Chassis for the robot:-** We need a body for our Robo which will be provided by the chassis.

12. **Rear and front wheels:**- To move the device smoothly.

13. **Jumper wires:-** To make the connections.

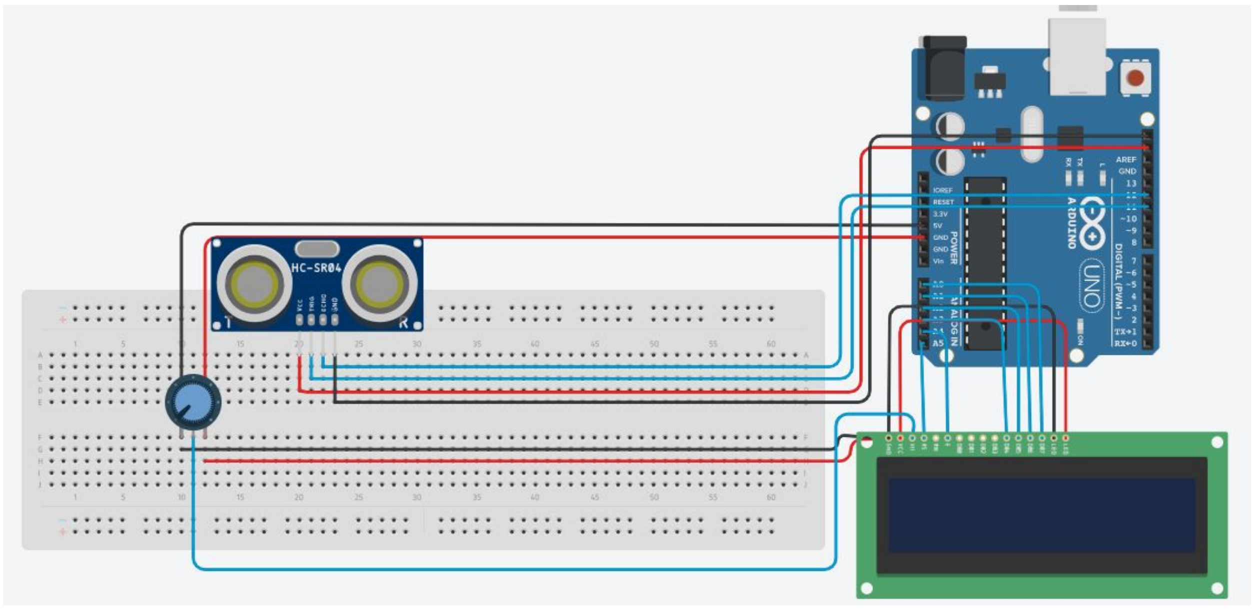
14. **Ultrasonic sensor:-** We want To identify the parking location, it is the eye of our Robo.

15. **Power supply(Battery):-** We want to provide the power to run the robot which will be provided by the 9-volt batteries, which are like the food of our device.

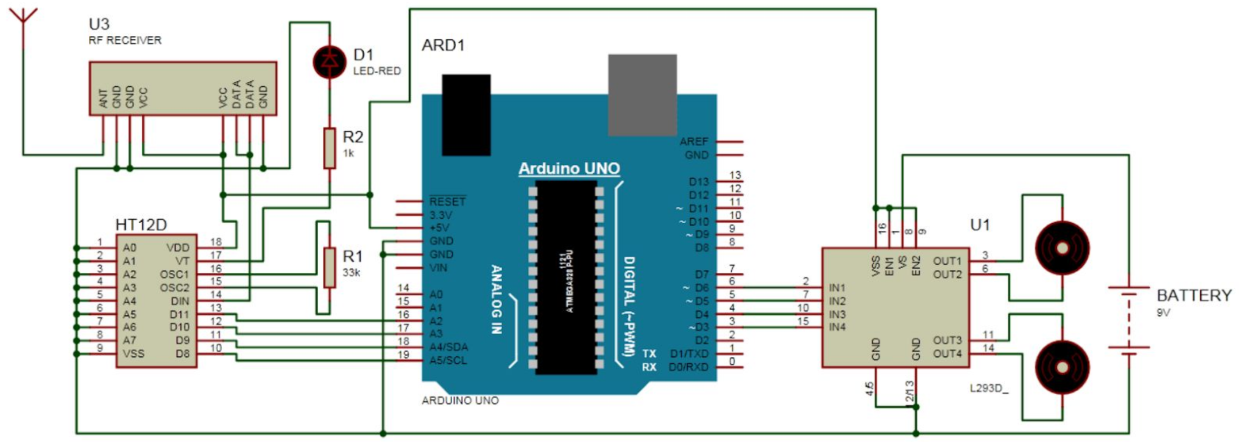
16. **Breadboard**:- To connect various circuits through jumper wires.

Circuit diagram

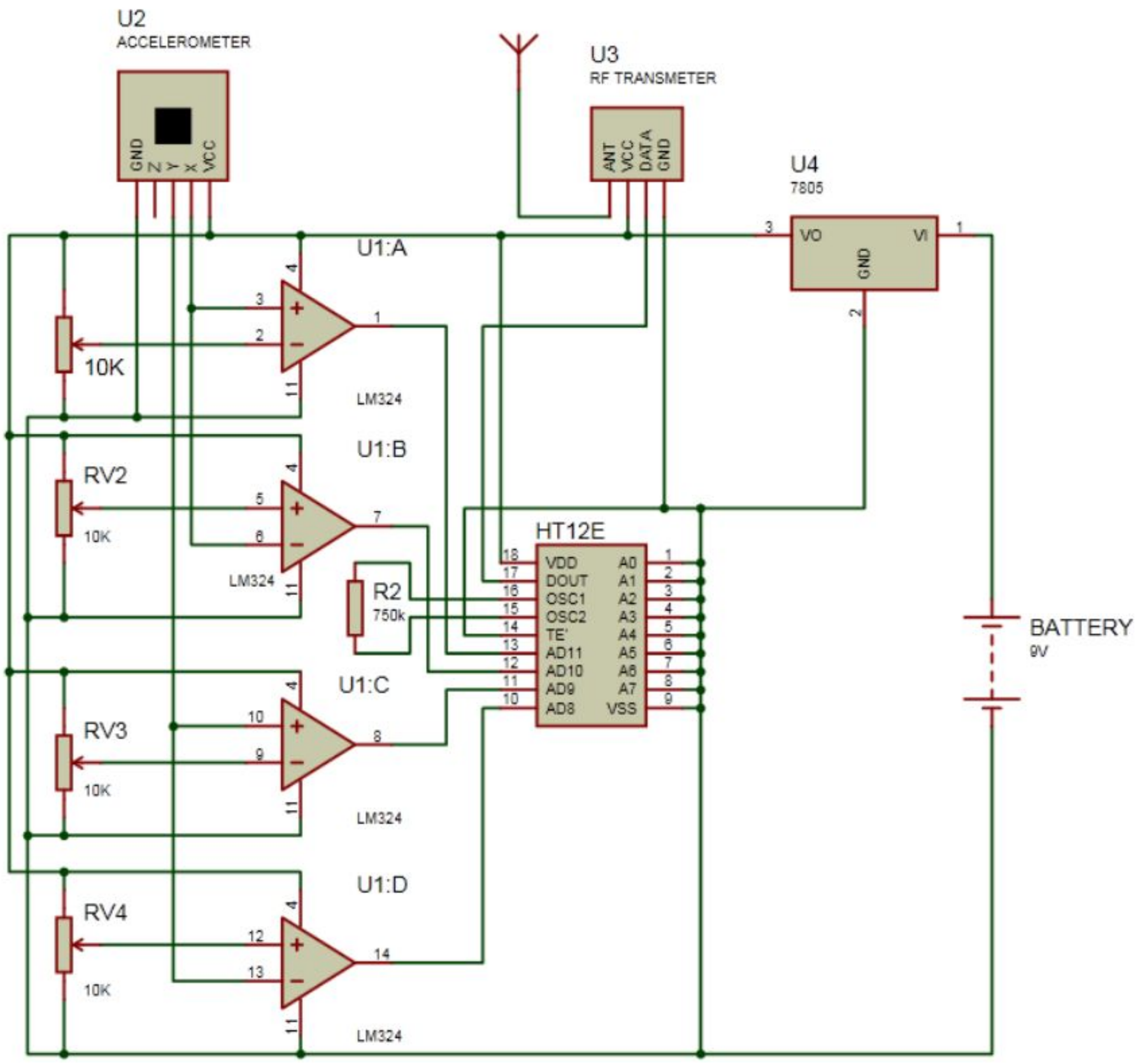
1. **Parking finder:-**



2. **Gesture receiver:-**

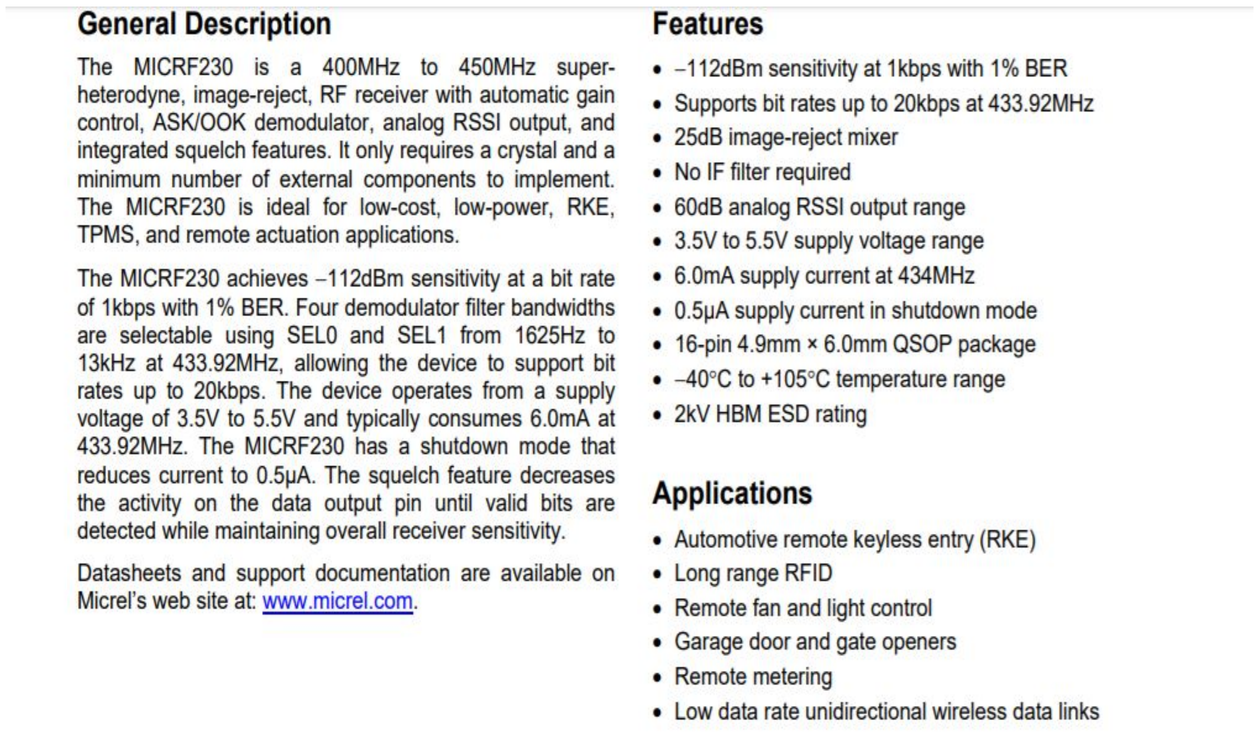


3. Remote:-


U2
ACCELEROMETER
GND Z Y X VCC
U3
RF TRANSMETER
ANT VCC DATA GND
U4
7805
VO VI GND
U1:A
U1:B
U1:C
U1:D
LM324
10K
RV2
RV3
RV4
R2
750k
HT12E
VDD DOUT OSC1 OSC2 TE' AD11 AD10 AD9 AD8
A0 A1 A2 A3 A4 A5 A6 A7 VSS
BATTERY
9V

Datasheet of major components

1. **434MHz RF Receiver:-**



General Description

The MICRF230 is a 400MHz to 450MHz super-heterodyne, image-reject, RF receiver with automatic gain control, ASK/OOK demodulator, analog RSSI output, and integrated squelch features. It only requires a crystal and a minimum number of external components to implement. The MICRF230 is ideal for low-cost, low-power, RKE, TPMS, and remote actuation applications.

The MICRF230 achieves −112dBm sensitivity at a bit rate of 1kbps with 1% BER. Four demodulator filter bandwidths are selectable using SEL0 and SEL1 from 1625Hz to 13kHz at 433.92MHz, allowing the device to support bit rates up to 20kbps. The device operates from a supply voltage of 3.5V to 5.5V and typically consumes 6.0mA at 433.92MHz. The MICRF230 has a shutdown mode that reduces current to 0.5µA. The squelch feature decreases the activity on the data output pin until valid bits are detected while maintaining overall receiver sensitivity.

Datasheets and support documentation are available on Micrel's web site at: www.micrel.com.

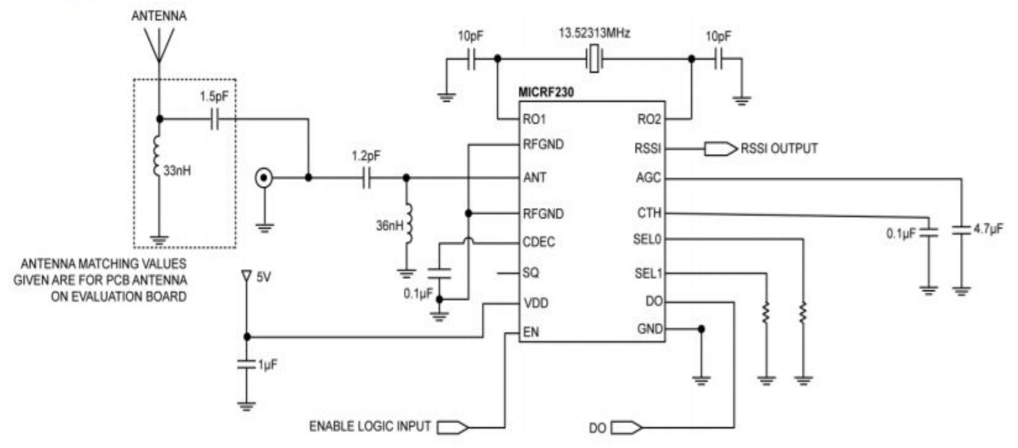
Features

- −112dBm sensitivity at 1kbps with 1% BER
- Supports bit rates up to 20kbps at 433.92MHz
- 25dB image-reject mixer
- No IF filter required
- 60dB analog RSSI output range
- 3.5V to 5.5V supply voltage range
- 6.0mA supply current at 434MHz
- 0.5µA supply current in shutdown mode
- 16-pin 4.9mm × 6.0mm QSOP package
- −40°C to +105°C temperature range
- 2kV HBM ESD rating

Applications

- Automotive remote keyless entry (RKE)
- Long range RFID
- Remote fan and light control
- Garage door and gate openers
- Remote metering
- Low data rate unidirectional wireless data links

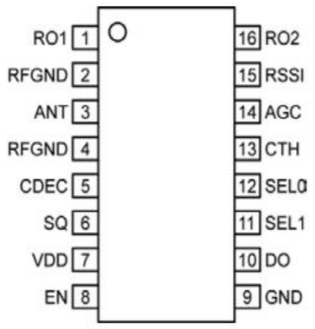
Typical Application



MICRF230 Typical Application Circuit for 433.92MHz

Part Number	Top Marking	Junction Temperature Range	Package
MICRF230YQS	MICRF230YQS	–40°C to +105°C	16-Pin 4.9mm × 6.0mm QSOP

Pin Configuration



16-Pin 4.9mm × 6.0mm QSOP (QS)
(Top View)

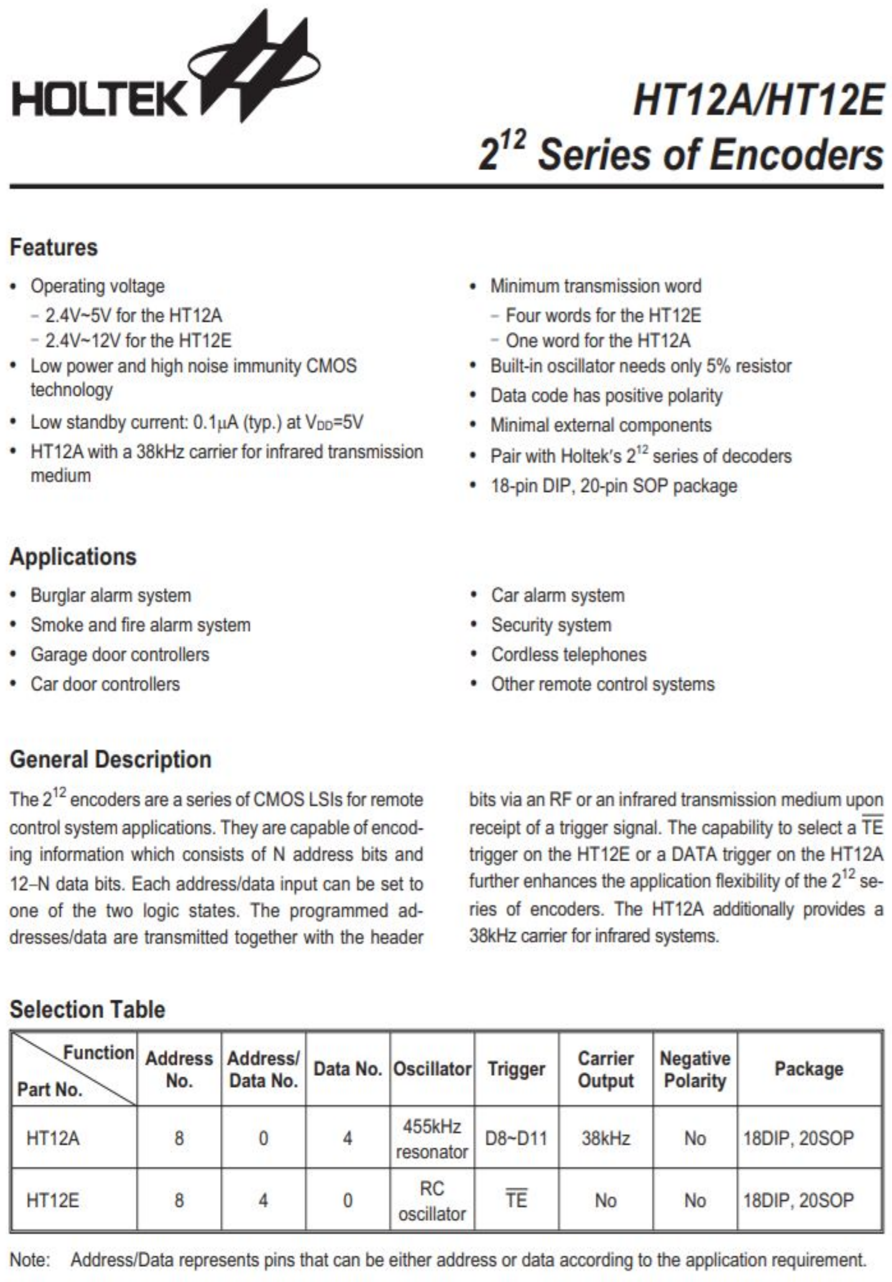
Pin Description

Pin Number	Pin Name	Type	Pin Function
1	RO1	Input	Reference resonator connection (to the Pierce oscillator). Can also be driven by external reference signal of 200mV$_{P-P}$ to 1.5V$_{P-P}$ amplitude maximum. Internal capacitance of 7pF to GND during normal operation.
2	RFGND	Supply	Ground connection for ANT RF input. Connect to PCB ground plane.
3	ANT	Input	Antenna input. RF signal input from antenna. Internally AC coupled. It is recommended to use a matching network with an inductor to RF ground to improve ESD protection.
4	RFGND	Supply	Ground connection for ANT RF input. Connect to PCB ground plane.
5	CDEC	Supply	Internal supply decoupling access. Bypass to PCB ground plane with a 0.1µF ceramic capacitor located as close to pin as possible. Maximum operating voltage is 3.6V.
6	SQ	Input	Squelch control logic-level input. An internal pull-up (3µA typical) pulls the logic-input HIGH when the device is enabled. A logic LOW on SQ squelches, or reduces, the random activity on DO pin when there is no RF input signal.
7	VDD	Supply	Positive supply connection (for all chip functions). Bypass with 1µF capacitor located as close to the VDD pin as possible.
8	EN	Input	Enable control logic-level input. A logic-level HIGH enable the device. A logic-level LOW put the device to shutdown mode. An internal pull-down (3µA typical) pulls the logic input LOW. The device is designed to start up in shutdown state. The EN pin should be kept at logic low (shutdown state) until after the supply voltage on VDD is stabilized. If the application is designed to have the EN pin always pulled high, it is recommended to add a shunt capacitor of 0.47µF from the EN pin to ground.
9	GND	Supply	Ground connection for all chip functions except for RF input. Connect to PCB ground plane.

Pin Description (Continued)

Pin Number	Pin Name	Type	Pin Function
10	DO	Output	Demodulation data output. A current limited CMOS output in normal operation. An internal pull-down of 25kΩ is present when device is in shutdown.
11	SEL1	Input	Logic control input with active internal pull-up (3µA typical). It can be used to select the low-pass filter bandwidth in the absence register control (Table 1).
12	SEL0	Input	Logic control input with active internal pull-up (3µA typical). It can be used to select the low-pass filter bandwidth in the absence register control (Table 1).
13	CTH	Input/Output	Demodulation threshold voltage integration capacitor. Capacitor to GND sets the settling time for the demodulation data slice level. Values above 1nF are recommended and should be optimized for data rate and data profile. Connect a 0.1µF capacitor from CTH pin to GND to provide a stable slicing threshold.
14	AGC	Input/Output	AGC filter capacitor connection. Connect a capacitor from this pin to GND. Refer to the "AGC Loop" in the *Receiver Operation* section for information on the capacitor value.
15	RSSI	Output	Received Signal Strength Indicator output. The voltage on this pin is an inversed amplified version of the voltage on AGC. Output is from a buffer with typically 200Ω output impedance.
16	RO2	Output	Pierce Oscillator Output for Crystal Output: Internal capacitance of 7pF to GND during normal operation.

2. HT12E (Encoder IC):-



HOLTEK

HT12A/HT12E

2^{12} Series of Encoders

Features

- Operating voltage
  - 2.4V~5V for the HT12A
  - 2.4V~12V for the HT12E
- Low power and high noise immunity CMOS technology
- Low standby current: 0.1µA (typ.) at V_{DD}=5V
- HT12A with a 38kHz carrier for infrared transmission medium
- Minimum transmission word
  - Four words for the HT12E
  - One word for the HT12A
- Built-in oscillator needs only 5% resistor
- Data code has positive polarity
- Minimal external components
- Pair with Holtek's 2^{12} series of decoders
- 18-pin DIP, 20-pin SOP package

Applications

- Burglar alarm system
- Smoke and fire alarm system
- Garage door controllers
- Car door controllers
- Car alarm system
- Security system
- Cordless telephones
- Other remote control systems

General Description

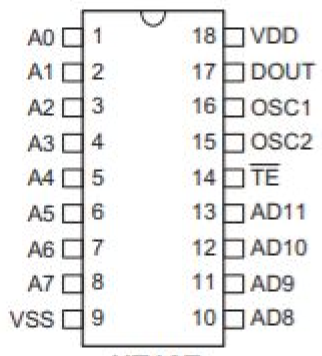
The 2^{12} encoders are a series of CMOS LSIs for remote control system applications. They are capable of encoding information which consists of N address bits and 12–N data bits. Each address/data input can be set to one of the two logic states. The programmed addresses/data are transmitted together with the header bits via an RF or an infrared transmission medium upon receipt of a trigger signal. The capability to select a $\overline{TE}$ trigger on the HT12E or a DATA trigger on the HT12A further enhances the application flexibility of the 2^{12} series of encoders. The HT12A additionally provides a 38kHz carrier for infrared systems.

Selection Table

Function / Part No.	Address No.	Address/ Data No.	Data No.	Oscillator	Trigger	Carrier Output	Negative Polarity	Package
HT12A	8	0	4	455kHz resonator	D8~D11	38kHz	No	18DIP, 20SOP
HT12E	8	4	0	RC oscillator	$\overline{TE}$	No	No	18DIP, 20SOP

Note: Address/Data represents pins that can be either address or data according to the application requirement.

8-Address
4-Address/Data

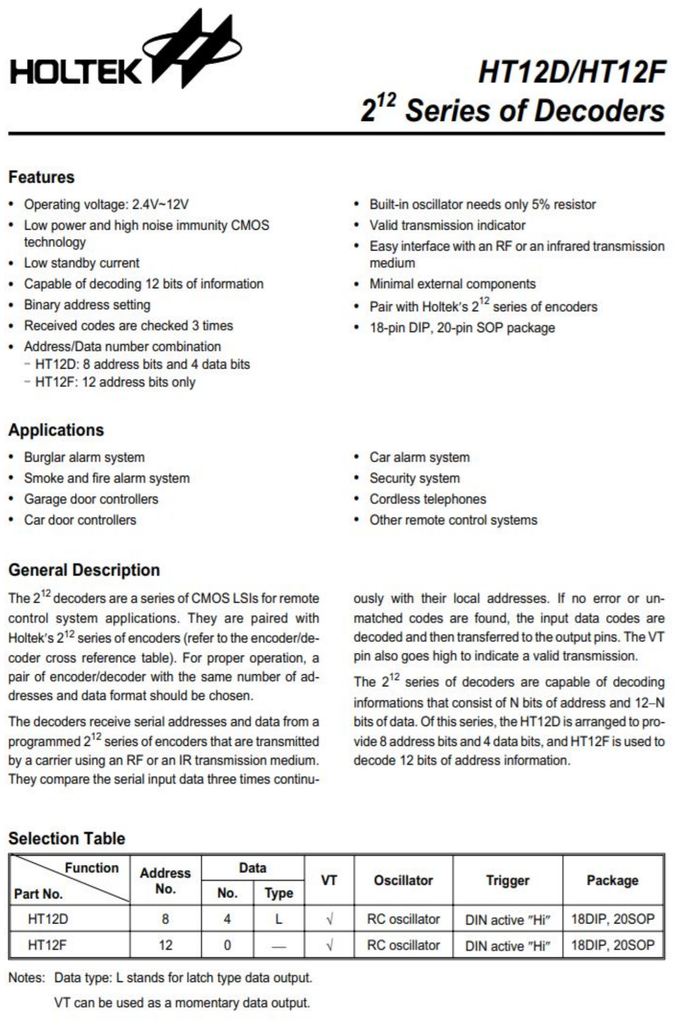


HT12E
–18 DIP-A

HT12E Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD}	Operating Voltage	—	—	2.4	5	12	V
I_{STB}	Standby Current	3V	Oscillator stops	—	0.1	1	μA
		12V		—	2	4	μA
I_{DD}	Operating Current	3V	No load, f_{OSC}=3kHz	—	40	80	μA
		12V		—	150	300	μA
I_{DOUT}	Output Drive Current	5V	V_{OH}=0.9V_{DD} (Source)	–1	–1.6	—	mA
			V_{OL}=0.1V_{DD} (Sink)	1	1.6	—	mA
V_{IH}	"H" Input Voltage	—	—	0.8V_{DD}	—	V_{DD}	V
V_{IL}	"L" Input Voltage	—	—	0	—	0.2V_{DD}	V
f_{OSC}	Oscillator Frequency	5V	R_{OSC}=1.1MΩ	—	3	—	kHz
$R_{\overline{TE}}$	$\overline{TE}$ Pull-high Resistance	5V	$V_{\overline{TE}}$=0V	—	1.5	3	MΩ

3. HT-12D(decoder IC):-



HOLTEK

HT12D/HT12F
2^{12} Series of Decoders

Features

- Operating voltage: 2.4V~12V
- Low power and high noise immunity CMOS technology
- Low standby current
- Capable of decoding 12 bits of information
- Binary address setting
- Received codes are checked 3 times
- Address/Data number combination
  - HT12D: 8 address bits and 4 data bits
  - HT12F: 12 address bits only
- Built-in oscillator needs only 5% resistor
- Valid transmission indicator
- Easy interface with an RF or an infrared transmission medium
- Minimal external components
- Pair with Holtek's 2^{12} series of encoders
- 18-pin DIP, 20-pin SOP package

Applications

- Burglar alarm system
- Smoke and fire alarm system
- Garage door controllers
- Car door controllers
- Car alarm system
- Security system
- Cordless telephones
- Other remote control systems

General Description

The 2^{12} decoders are a series of CMOS LSIs for remote control system applications. They are paired with Holtek's 2^{12} series of encoders (refer to the encoder/decoder cross reference table). For proper operation, a pair of encoder/decoder with the same number of addresses and data format should be chosen.

The decoders receive serial addresses and data from a programmed 2^{12} series of encoders that are transmitted by a carrier using an RF or an IR transmission medium. They compare the serial input data three times continuously with their local addresses. If no error or unmatched codes are found, the input data codes are decoded and then transferred to the output pins. The VT pin also goes high to indicate a valid transmission.

The 2^{12} series of decoders are capable of decoding informations that consist of N bits of address and 12–N bits of data. Of this series, the HT12D is arranged to provide 8 address bits and 4 data bits, and HT12F is used to decode 12 bits of address information.

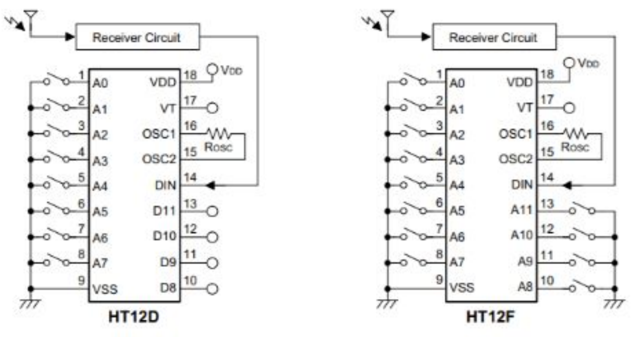
Selection Table

Function / Part No.	Address No.	Data		VT	Oscillator	Trigger	Package
		No.	Type				
HT12D	8	4	L	√	RC oscillator	DIN active "Hi"	18DIP, 20SOP
HT12F	12	0	—	√	RC oscillator	DIN active "Hi"	18DIP, 20SOP

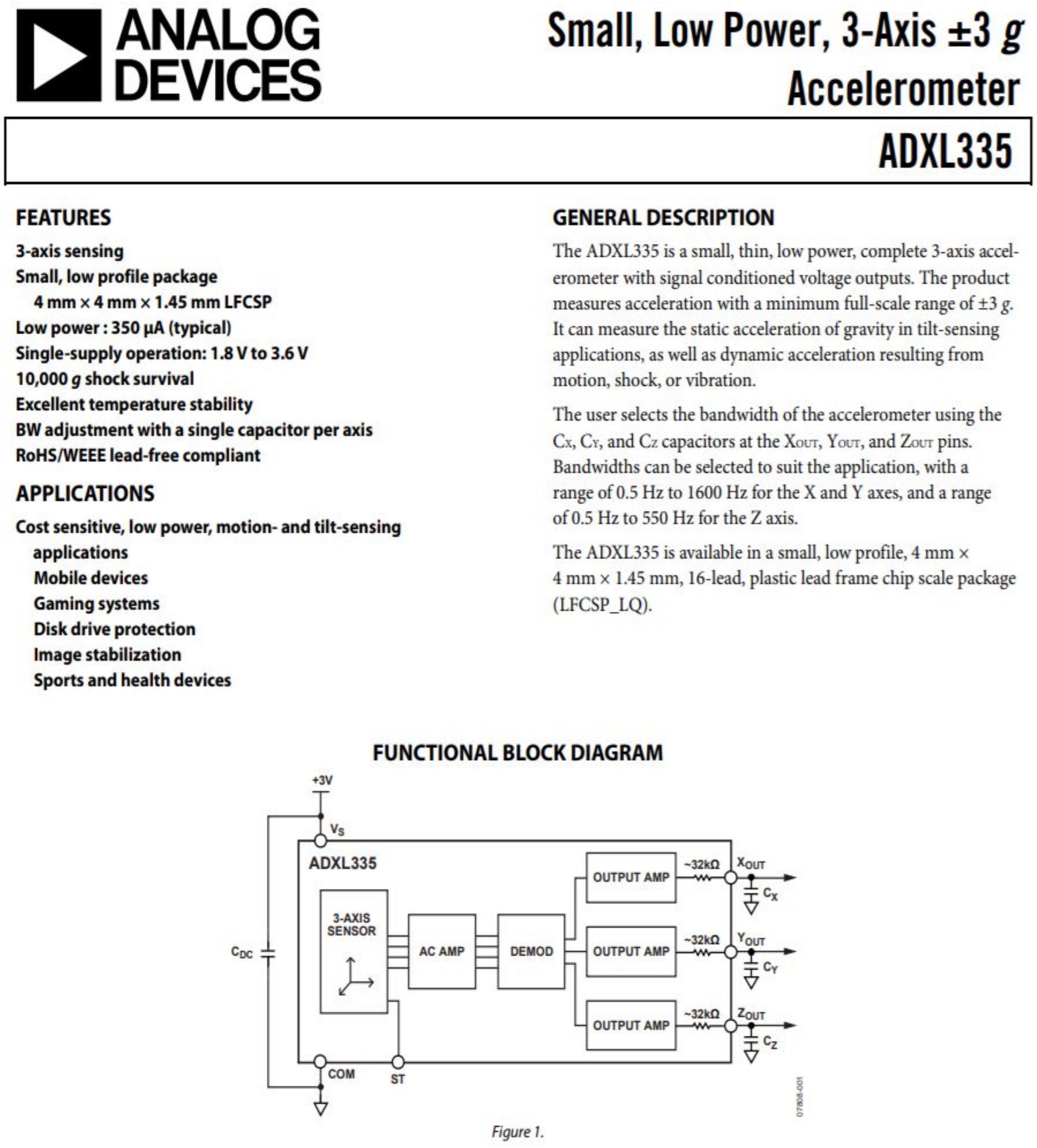
Notes: Data type: L stands for latch type data output.
VT can be used as a momentary data output.

HOLTEK **HT12D/HT12F**

Application Circuits



4. **Accelerometer (ADXL335):-**



ANALOG DEVICES

Small, Low Power, 3-Axis ±3 *g* Accelerometer

ADXL335

FEATURES

3-axis sensing
Small, low profile package
  4 mm × 4 mm × 1.45 mm LFCSP
Low power : 350 µA (typical)
Single-supply operation: 1.8 V to 3.6 V
10,000 *g* shock survival
Excellent temperature stability
BW adjustment with a single capacitor per axis
RoHS/WEEE lead-free compliant

APPLICATIONS

Cost sensitive, low power, motion- and tilt-sensing
  applications
  Mobile devices
  Gaming systems
  Disk drive protection
  Image stabilization
  Sports and health devices

GENERAL DESCRIPTION

The ADXL335 is a small, thin, low power, complete 3-axis accelerometer with signal conditioned voltage outputs. The product measures acceleration with a minimum full-scale range of ±3 *g*. It can measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration.

The user selects the bandwidth of the accelerometer using the C_X, C_Y, and C_Z capacitors at the X_{OUT}, Y_{OUT}, and Z_{OUT} pins. Bandwidths can be selected to suit the application, with a range of 0.5 Hz to 1600 Hz for the X and Y axes, and a range of 0.5 Hz to 550 Hz for the Z axis.

The ADXL335 is available in a small, low profile, 4 mm × 4 mm × 1.45 mm, 16-lead, plastic lead frame chip scale package (LFCSP_LQ).

FUNCTIONAL BLOCK DIAGRAM

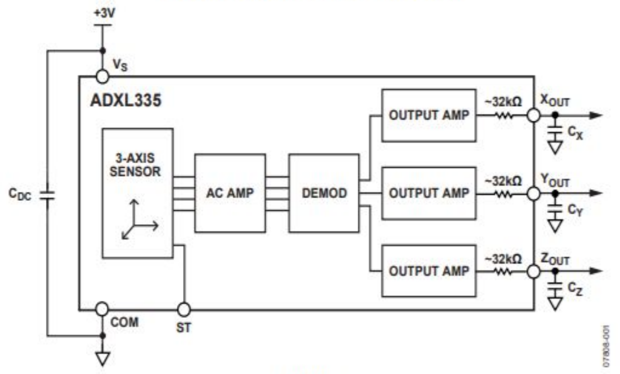


Figure 1.

SPECIFICATIONS

$T_A = 25°C$, $V_S = 3$ V, $C_X = C_Y = C_Z = 0.1$ µF, acceleration = 0 *g*, unless otherwise noted. All minimum and maximum specifications are guaranteed. Typical specifications are not guaranteed.

Table 1.

Parameter	Conditions	Min	Typ	Max	Unit
SENSOR INPUT	Each axis				
Measurement Range		±3	±3.6		*g*
Nonlinearity	% of full scale		±0.3		%
Package Alignment Error			±1		Degrees
Interaxis Alignment Error			±0.1		Degrees
Cross-Axis Sensitivity[1]			±1		%
SENSITIVITY (RATIOMETRIC)[2]	Each axis				
Sensitivity at X_{OUT}, Y_{OUT}, Z_{OUT}	$V_S = 3$ V	270	300	330	mV/*g*
Sensitivity Change Due to Temperature[3]	$V_S = 3$ V		±0.01		%/°C
ZERO *g* BIAS LEVEL (RATIOMETRIC)					
0 *g* Voltage at X_{OUT}, Y_{OUT}	$V_S = 3$ V	1.35	1.5	1.65	V
0 *g* Voltage at Z_{OUT}	$V_S = 3$ V	1.2	1.5	1.8	V
0 *g* Offset vs. Temperature			±1		m*g*/°C
NOISE PERFORMANCE					
Noise Density X_{OUT}, Y_{OUT}			150		µ*g*/√Hz rms
Noise Density Z_{OUT}			300		µ*g*/√Hz rms
FREQUENCY RESPONSE[4]					
Bandwidth X_{OUT}, Y_{OUT}[5]	No external filter		1600		Hz
Bandwidth Z_{OUT}[5]	No external filter		550		Hz
R_{FILT} Tolerance			32 ± 15%		kΩ
Sensor Resonant Frequency			5.5		kHz
SELF-TEST[6]					
Logic Input Low			+0.6		V
Logic Input High			+2.4		V
ST Actuation Current			+60		µA
Output Change at X_{OUT}	Self-Test 0 to Self-Test 1	−150	−325	−600	mV
Output Change at Y_{OUT}	Self-Test 0 to Self-Test 1	+150	+325	+600	mV
Output Change at Z_{OUT}	Self-Test 0 to Self-Test 1	+150	+550	+1000	mV
OUTPUT AMPLIFIER					
Output Swing Low	No load		0.1		V
Output Swing High	No load		2.8		V
POWER SUPPLY					
Operating Voltage Range		1.8		3.6	V
Supply Current	$V_S = 3$ V		350		µA
Turn-On Time[7]	No external filter		1		ms
TEMPERATURE					
Operating Temperature Range		−40		+85	°C

[1] Defined as coupling between any two axes.
[2] Sensitivity is essentially ratiometric to V_S.
[3] Defined as the output change from ambient-to-maximum temperature or ambient-to-minimum temperature.
[4] Actual frequency response controlled by user-supplied external filter capacitors (C_X, C_Y, C_Z).
[5] Bandwidth with external capacitors = 1/(2 × π × 32 kΩ × C). For C_X, C_Y = 0.003 µF, bandwidth = 1.6 kHz. For C_Z = 0.01 µF, bandwidth = 500 Hz. For C_X, C_Y, C_Z = 10 µF,

PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

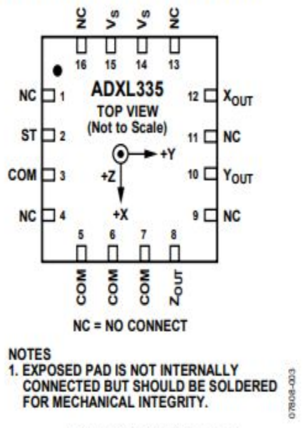


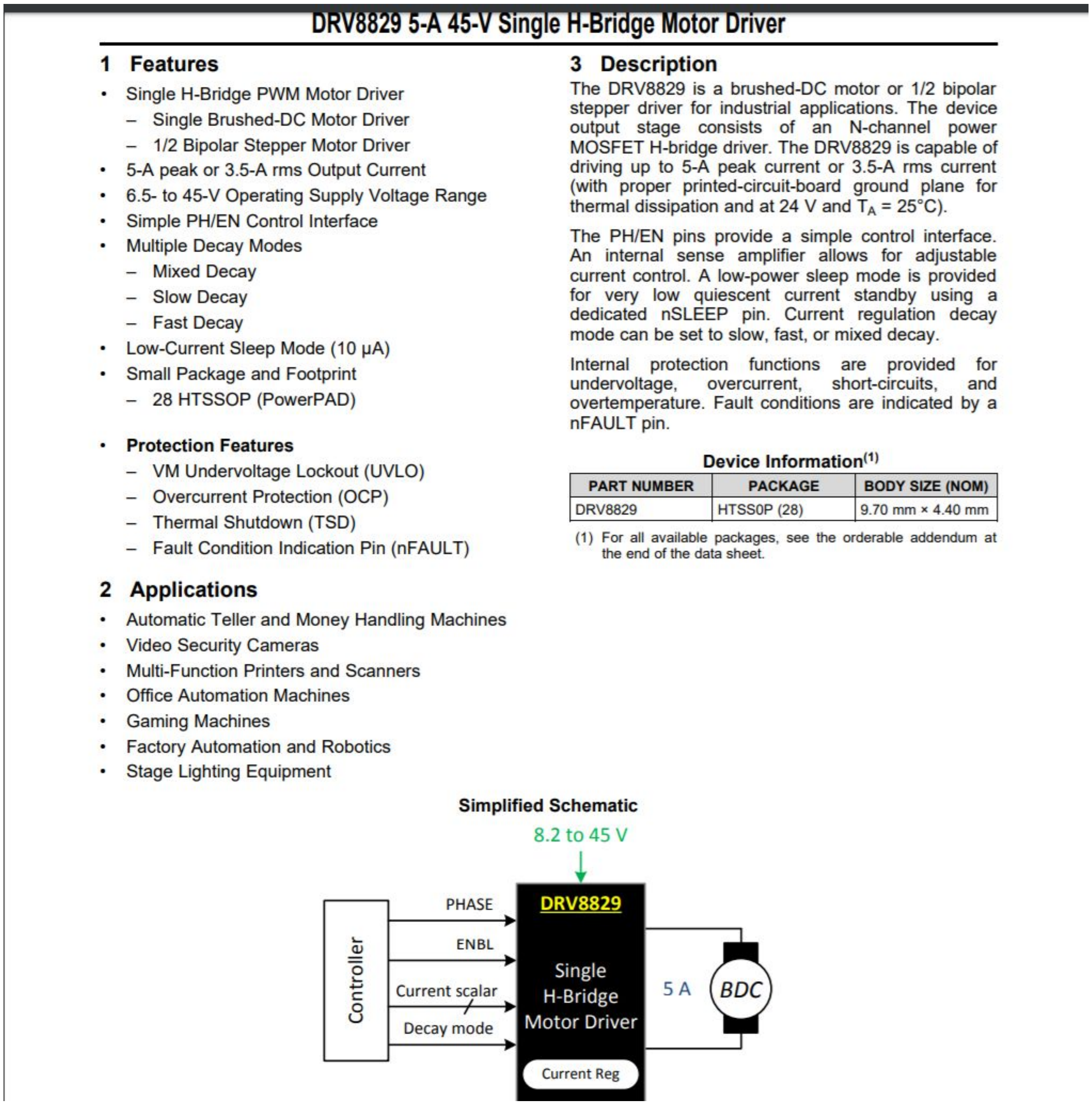
Figure 2. Pin Configuration

Table 3. Pin Function Descriptions

Pin No.	Mnemonic	Description
1	NC	No Connect[1].
2	ST	Self-Test.
3	COM	Common.
4	NC	No Connect[1].
5	COM	Common.
6	COM	Common.
7	COM	Common.
8	Z_{OUT}	Z Channel Output.
9	NC	No Connect[1].
10	Y_{OUT}	Y Channel Output.
11	NC	No Connect[1].
12	X_{OUT}	X Channel Output.
13	NC	No Connect[1].
14	V_S	Supply Voltage (1.8 V to 3.6 V).
15	V_S	Supply Voltage (1.8 V to 3.6 V).
16	NC	No Connect[1].
EP	Exposed Pad	Not internally connected. Solder for mechanical integrity.

[1]NC pins are not internally connected and can be tied to COM pins, unless otherwise noted.

5. H-bridge



DRV8829 5-A 45-V Single H-Bridge Motor Driver

1 Features

- Single H-Bridge PWM Motor Driver
  - Single Brushed-DC Motor Driver
  - 1/2 Bipolar Stepper Motor Driver
- 5-A peak or 3.5-A rms Output Current
- 6.5- to 45-V Operating Supply Voltage Range
- Simple PH/EN Control Interface
- Multiple Decay Modes
  - Mixed Decay
  - Slow Decay
  - Fast Decay
- Low-Current Sleep Mode (10 µA)
- Small Package and Footprint
  - 28 HTSSOP (PowerPAD)
- **Protection Features**
  - VM Undervoltage Lockout (UVLO)
  - Overcurrent Protection (OCP)
  - Thermal Shutdown (TSD)
  - Fault Condition Indication Pin (nFAULT)

2 Applications

- Automatic Teller and Money Handling Machines
- Video Security Cameras
- Multi-Function Printers and Scanners
- Office Automation Machines
- Gaming Machines
- Factory Automation and Robotics
- Stage Lighting Equipment

3 Description

The DRV8829 is a brushed-DC motor or 1/2 bipolar stepper driver for industrial applications. The device output stage consists of an N-channel power MOSFET H-bridge driver. The DRV8829 is capable of driving up to 5-A peak current or 3.5-A rms current (with proper printed-circuit-board ground plane for thermal dissipation and at 24 V and T_A = 25°C).

The PH/EN pins provide a simple control interface. An internal sense amplifier allows for adjustable current control. A low-power sleep mode is provided for very low quiescent current standby using a dedicated nSLEEP pin. Current regulation decay mode can be set to slow, fast, or mixed decay.

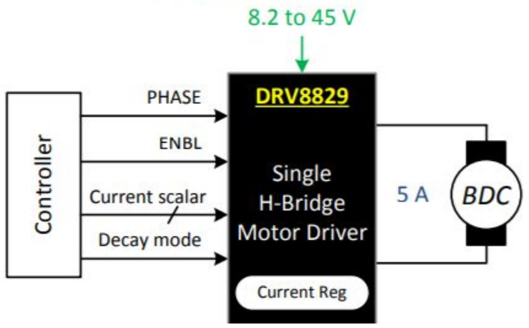
Internal protection functions are provided for undervoltage, overcurrent, short-circuits, and overtemperature. Fault conditions are indicated by a nFAULT pin.

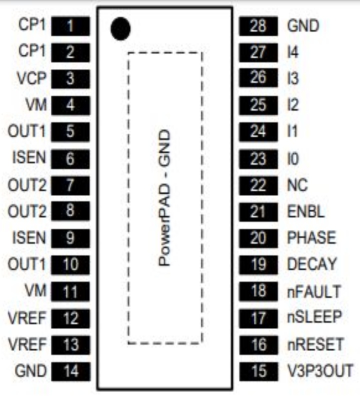
Device Information(1)

PART NUMBER	PACKAGE	BODY SIZE (NOM)
DRV8829	HTSS0P (28)	9.70 mm × 4.40 mm

(1) For all available packages, see the orderable addendum at the end of the data sheet.

Simplified Schematic





Pin Functions

PIN		I/O(1)	DESCRIPTION	EXTERNAL COMPONENTS OR CONNECTIONS
NAME	**NO.**			
POWER AND GROUND				
GND	14, 28	—	Device ground	
VM	4, 11	—	Bridge power supply	Connect to motor supply (8.2 V to 45 V). Both pins must be connected to same supply.
V3P3OUT	15	O	3.3-V regulator output	Bypass to GND with a 0.47-μF to 6.3-V ceramic capacitor. Can be used to supply VREF.
CP1	1	IO	Charge pump flying capacitor	Connect a 0.01-μF to 50-V capacitor between CP1 and CP2.
CP2	2	IO	Charge pump flying capacitor	
VCP	3	IO	High-side gate drive voltage	Connect a 0.1-μF to 16-V ceramic capacitor and 1-MΩ resistor to VM.
CONTROL				
ENBL	21	I	Bridge enable	Logic high to enable H-bridge. Internal pulldown.
PHASE	20	I	Bridge phase (direction)	Logic high sets OUT1 high, OUT2 low. Internal pulldown.
I0	23	I	Current set inputs	Sets winding current as a percentage of full-scale. Internal pulldown.
I1	24	I		
I2	25	I		
I3	26	I		
I4	27	I		
DECAY	19	I	Decay mode	Low = slow decay, open = mixed decay, high = fast decay Internal pulldown and pullup.
nRESET	16	I	Reset input	Active-low reset input initializes internal logic and disables the H-bridge outputs. Internal pulldown.
nSLEEP	17	I	Sleep mode input	Logic high to enable device, logic low to enter low-power sleep mode. Internal pulldown.
VREF	12, 13	I	Current set reference input	Reference voltage for winding current set. Both pins must be connected together on the PCB.
STATUS				
nFAULT	18	OD	Fault	Logic low when in fault condition (overtemperature, overcurrent)

(1) Directions: I = input, O = output, OZ = tri-state output, OD = open-drain output, IO = input/output

6. Ultrasonic sensor:-

HC-SR04 User Guide

Part 1 Ultrasonic Introduction

1. 1 Ultrasonic Definition

The human ear can hear sound frequency around 20HZ ~ 20KHZ, and ultrasonic is the sound wave beyond the human ability of 20KHZ .

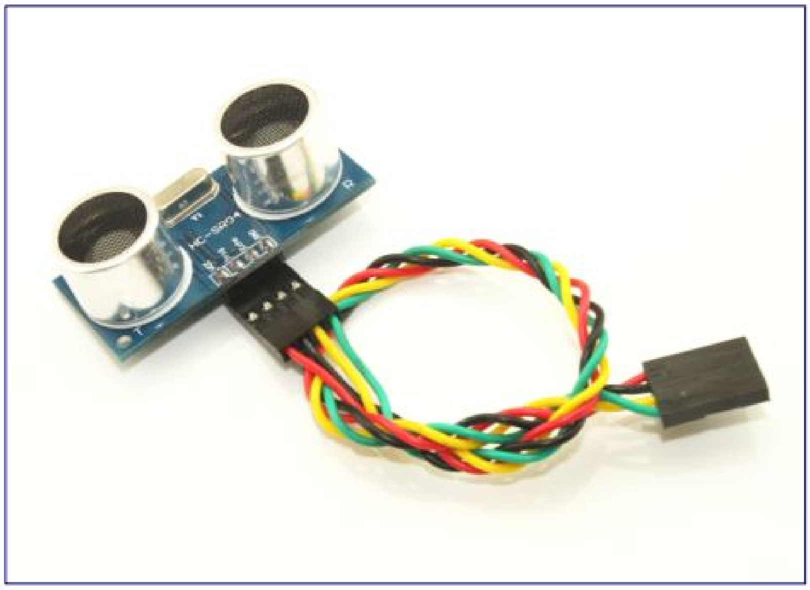
1.2 Ultrasonic distance measurement principle

Ultrasonic transmitter emitted an ultrasonic wave in one direction, and started timing when it launched. Ultrasonic spread in the air, and would return immediately when it encountered obstacles on the way. At last, the ultrasonic receiver would stop timing when it received the reflected wave. As Ultrasonic spread velocity is 340m / s in the air, based on the timer record **t**, we can calculate the distance (s) between the obstacle and transmitter, namely: s = 340t / 2, which is so- called time difference distance measurement principle

The principle of ultrasonic distance measurement used the already-known air spreading velocity, measuring the time from launch to reflection when it encountered obstacle, and then calculate the distance between the transmitter and the obstacle according to the time and the velocity. Thus, the principle of ultrasonic distance measurement is the same with radar.

Distance Measurement formula is expressed as: L = C X T

In the formula, L is the measured distance, and C is the ultrasonic spreading velocity in air, also, T represents time (T is half the time value from transmitting to receiving).



2.3、Module pin definitions

Types	Pin Symbol	Pin Function Description
HC-SR04	VCC	5V power supply
	Trig	Trigger pin
	Echo	Receive pin
	GND	Power ground

2.4、Electrical parameters

Electrical Parameters	HC-SR04 Ultrasonic Module
Operating Voltage	DC-5V
Operating Current	15mA
Operating Frequency	40KHZ
Farthest Range	4m
Nearest Range	2cm
Measuring Angle	15 Degree
Input Trigger Signal	10us TTL pulse
Output Echo Signal	Output TTL level signal, proportional with range
Dimensions	45*20*15mm

7. LCD Display (GMD1621C):-

2 General information

Description

This is an LCD Display designed for E-blocks. It is a 16 character, 2-line alphanumeric LCD display connected to a single 9-way D-type connector. This allows the device to be connected to most E-Block I/O ports.

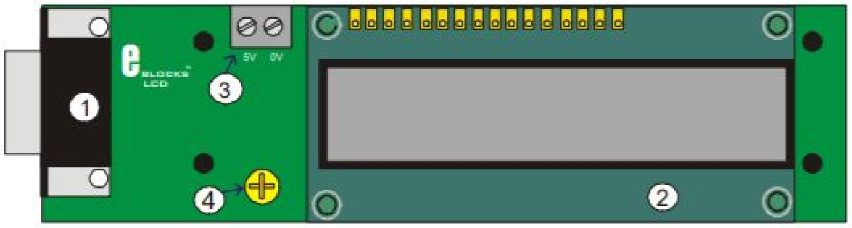
The LCD display requires data in a serial format, which is detailed in the user guide below. The display also requires a 5V power supply. Please take care not to exceed 5V, as this will cause damage to the device. The 5V is best generated from the E-blocks Multipogrammer or a 5V fixed regulated power supply.

The potentiometer RV1 is a contrast control that should be used to adjust the contrast of the display for the environment it is being used in.

Features

- E-blocks compatible
- Low cost
- Compatible with most I/O ports in the E-Block range (requires 5 I/O lines via 9 way D-type connector)
- Ease to develop programming code using Flowcode icons.

3 LCD Board Layout



1) 9 Way D-type Plug
2) 16 character, 2-line alphanumeric LCD display
3) 9 Screw terminal
4) Contrast Control

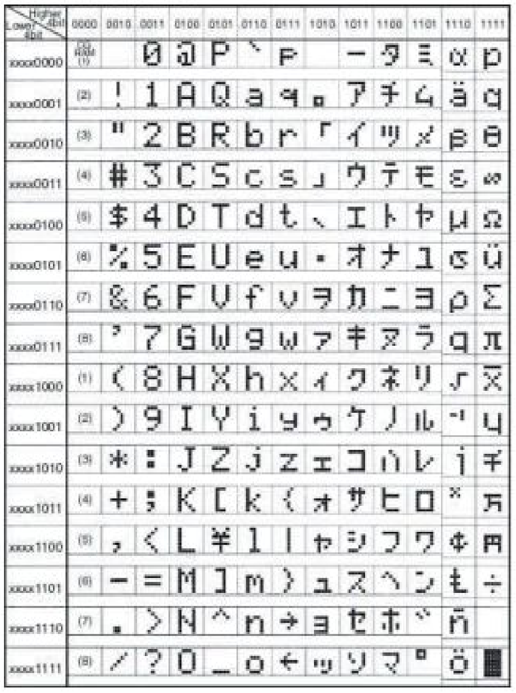
LCD Instruction Set

Instruction	Code					Description	Execution Time
MSB / LSB	B4	B3	B2	B1	B0		
Clear Display	0	0	0	0	0	Clear all display data. Set DDRAM address to 0. Move cursor to home position. Entry mode set to increment.	1.53 ms
		0	0	0	1		
Return Home	0	0	0	0	0	Set DDRAM address to 0. Move cursor to home position.	1.53 ms
		0	0	1	X		
Entry Mode Set	0	0	0	0	0	Sets cursor move direction (I/D), specifies to shift the display (S). These operations are performed during data read/write.	39 us
		0	1	I/D	SH		
Display Control	0	0	0	0	0	D is Display ON/OFF bit. C is Cursor ON/OFF bit. B is Blink Cursor ON/OFF bit.	39 us
		1	D	C	B		
Cursor/Display Shift	0	0	0	0	1	Sets cursor-move or display-shift (S/C). shift direction (R/L). DDRAM contents remains unchanged.	39 us
		S/C	R/L	X	X		
Function Set	0	0	0	1	0	Configuration data for setting up LCD. [Send First]	39 us
		1	0	X	X		
Set CGRAM Address	0	0	1	A5	A4	Sets the CGRAM address. CGRAM data is sent and received after this setting.	39 us
		A3	A2	A1	A0		
Set DDRAM Address	0	1	A6	A5	A4	Sets the DDRAM address. DDRAM data is sent and received after this setting.	39 us
		A3	A2	A1	A0		
Write Data to RAM	1	D7	D6	D5	D4	Writes data to CGRAM or DDRAM.	43 us
		D3	D2	D1	D0		

DDRAM is Display Data RAM
DDRAM address is location of cursor
CGRAM is Character Generator RAM
X is Don t Care

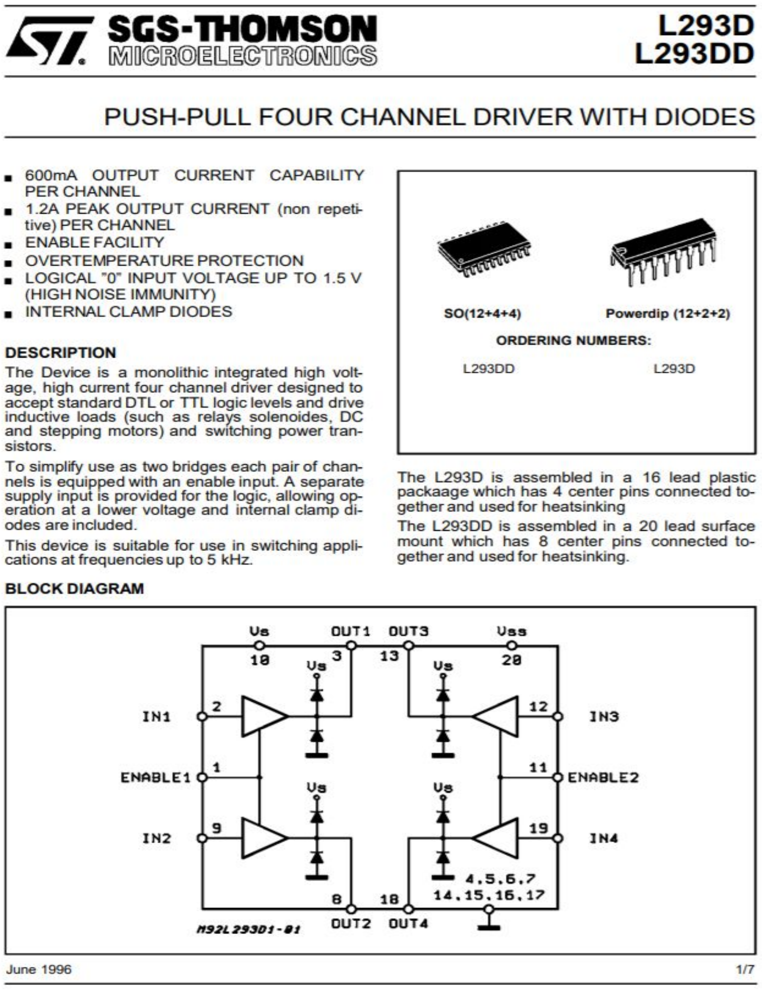
Bit Name	0	1
I/D	Decrement cursor position	Increment cursor position
SH	No display shift	Display shift
D	Display off	Display on
C	Cursor off	Cursor on
B	Cursor blink off	Cursor blink on
S/C	Move cursor	Shift display
R/L	Shift left	Shift right

LCD Character Set



Lower 4bit \ Higher 4bit	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)		0	@	P	`	p		ー	タ	ミ	α	p
xxxx0001	(2)	!	1	A	Q	a	q	。	ア	チ	ム	ä	q
xxxx0010	(3)	"	2	B	R	b	r	「	イ	ツ	メ	β	θ
xxxx0011	(4)	#	3	C	S	c	s	」	ウ	テ	モ	ε	∞
xxxx0100	(5)	$	4	D	T	d	t	、	エ	ト	ヤ	μ	Ω
xxxx0101	(6)	%	5	E	U	e	u	・	オ	ナ	ユ	σ	ü
xxxx0110	(7)	&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)	'	7	G	W	g	w	ァ	キ	ヌ	ラ	g	π
xxxx1000	(1)	(	8	H	X	h	x	ィ	ク	ネ	リ	√	x̄
xxxx1001	(2)	)	9	I	Y	i	y	ゥ	ケ	ノ	ル	⁻¹	y
xxxx1010	(3)	*	:	J	Z	j	z	ェ	コ	ハ	レ	j	千
xxxx1011	(4)	+	;	K	[	k	{	ォ	サ	ヒ	ロ	ˣ	万
xxxx1100	(5)	,	<	L	¥	l	|	ャ	シ	フ	ワ	¢	円
xxxx1101	(6)	-	=	M	]	m	}	ュ	ス	ヘ	ン	£	÷
xxxx1110	(7)	.	>	N	^	n	→	ョ	セ	ホ	゛	ñ	
xxxx1111	(8)	/	?	O	_	o	←	ッ	ソ	マ	゜	ö	█

8. L293D:-



SGS-THOMSON MICROELECTRONICS

L293D
L293DD

PUSH-PULL FOUR CHANNEL DRIVER WITH DIODES

- 600mA OUTPUT CURRENT CAPABILITY PER CHANNEL
- 1.2A PEAK OUTPUT CURRENT (non repetitive) PER CHANNEL
- ENABLE FACILITY
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)
- INTERNAL CLAMP DIODES

SO(12+4+4) Powerdip (12+2+2)

ORDERING NUMBERS:

L293DD L293D

DESCRIPTION

The Device is a monolithic integrated high voltage, high current four channel driver designed to accept standard DTL or TTL logic levels and drive inductive loads (such as relays solenoides, DC and stepping motors) and switching power transistors.

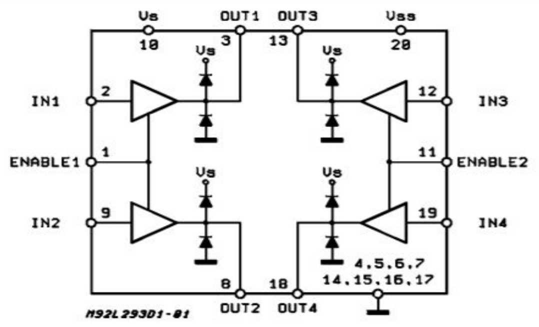
To simplify use as two bridges each pair of channels is equipped with an enable input. A separate supply input is provided for the logic, allowing operation at a lower voltage and internal clamp diodes are included.

This device is suitable for use in switching applications at frequencies up to 5 kHz.

The L293D is assembled in a 16 lead plastic packaage which has 4 center pins connected together and used for heatsinking

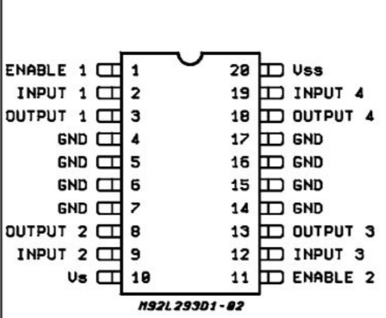
The L293DD is assembled in a 20 lead surface mount which has 8 center pins connected together and used for heatsinking.

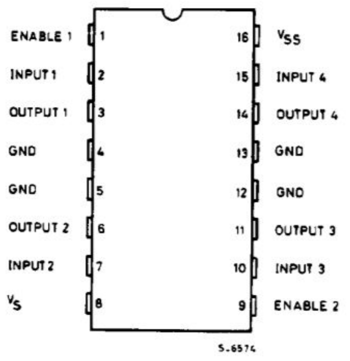
BLOCK DIAGRAM



June 1996 1/7

PIN CONNECTIONS (Top view)





SO(12+4+4) Powerdip(12+2+2)

9. 434 MHz RF Transmitter:-

SAW RESONATOR TRANSMITTER MODULE

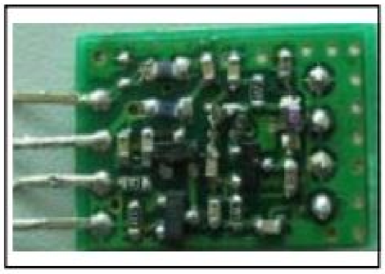
315/434 MHz ASK TRANSMITTER

Description

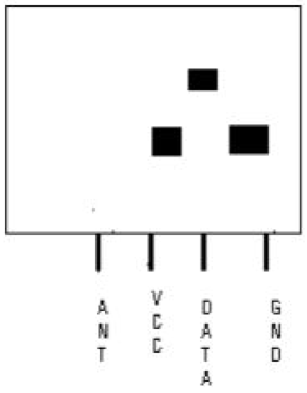
@433.92/315MHz Remote Keyless-Entry Transmitter.

@SAW RESONATOR

@ASK



The MO-SAWR is an ASK transmitter module .
The result is excellent performance in a simple-to-use .
The MO-SAWR is designed specifically for remote-control , wireless mouse and car alarm system operating at 433.92 in the USA under FCC Part 15 regulation.



Applications

- Car security system
- Remote keyless entry
- Garage door controller
- Home security
- Wireless mouse
- Automation system

Product Identification

315MHz	**MO-SAWR-AS315M**
433.92MHz	**MO-SAWR-AS434M**

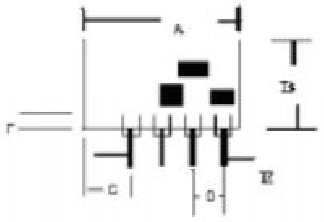
Absolute Maximum Ratings

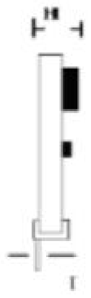
Parameter	Rating	Units
Supply Voltage	1.5—12.0	V DC
Operating Temperature	-20 to +85	℃

Absolute Maximum Ratings

Parameter	Symbol	Condition		Specification						Unit
				Min.	Typical				Max.	
Operation Voltage					1.5	3	5	12		V
Output power	Psens	DATA 5V 1Kbps Data Rate	315MHz		-11.8	4	10	16		dBm
			Supply current		3.1	11	20	57		mA
			434MHz		-8.5	4	10	16		dBm
			Supply current		2.9	11	22	59		mA
Tune on Time	Ton	Data start out by Vcc turn on		10	20					ms
Data Rate				200	1k				10k	bps
Input duty		Vcc=5V; 1kbps data rate		40					60	%

Pin Dimension

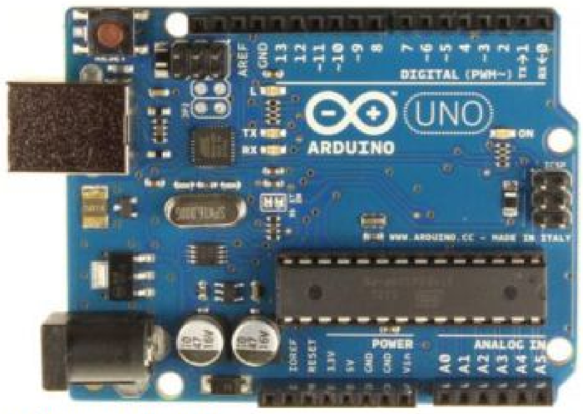


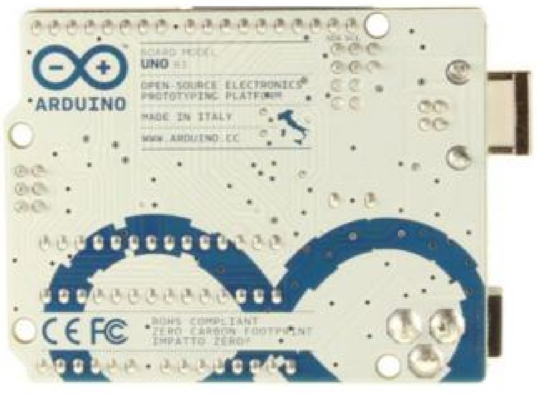


Dimensions	Millimeters	Dimensions	Millimeters
A	14 ± 0.25mm	F	2.50 ± 0.15mm
B	21 ± 0.25mm	G	3.50 ± 0.15mm
C	4.1 ± 0.30mm	H	5.5mm
D	2.54 ± 0.05mm	I	0.32 ± 0.05mm
E	0.65 ± 0.05mm		

10. Arduino UNO:-

Arduino Uno

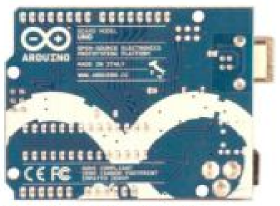

Arduino Uno R3 Front


Arduino Uno R3 Back

Arduino Uno R2 Front

Arduino Uno SMD

Arduino Uno Front


Arduino Uno Back

Overview

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.
The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the Atmega16U2 (Atmega8U2 up to version R2) programmed as a USB-to-serial converter.
Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.
Revision 3 of the board has the following new features:

- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Main gesture control code

```
#include &quot;Arduino.h&quot;
#define xPin A3
#define yPin A1
#define zPin A5

int GNDPin=A4; //Set Analog pin 4 as GND
int VccPin=A5; //Set Analog pin 5 as VCC

//int xPin=A3; //X axis input
//int yPin=A1; //Y axis input
//int zPin=A5; //Z axis input(not used)

int Q1=10,Q2=11,Q3=12,Q4=13; //Output pins to be connected to 10, 11, 12, 13 of Decoder IC

long x; //Variable for storing X coordinates
long y; //Variable for storing Y coordinates
long z; //Variable for storing Z coordinates

void setup(){
        Serial.begin(9600);
        pinMode(Q1,OUTPUT); //here Q1,Q2 is two side of motor1 and Q3,Q4 is two side of motor2
        pinMode(Q2,OUTPUT);
        pinMode(Q3,OUTPUT);
        pinMode(Q4,OUTPUT);
        pinMode(GNDPin, OUTPUT);
        pinMode(VccPin, OUTPUT);
        digitalWrite(GNDPin, LOW); //Set A4 pin LOW
        digitalWrite(VccPin, HIGH); //Set A5 pin HIGH
}

void loop(){
        x = analogRead(xPin); //Reads X coordinates
        y = analogRead(yPin); //Reads Y coordinates
        z = analogRead(zPin); //Reads Z coordinates (Not Used)
if(x&lt;340) // Change the value for adjusting sensitivity
        forward();
else if(x&gt;400) // Change the value for adjusting sensitivity
        backward();
else if(y&gt;400) // Change the value for adjusting sensitivity
        right();
else if(y&lt;340) // Change the value for adjusting sensitivity
        left();
Else
        stop_();
}

void stop_(){ // for stoping the boat we need to stop all motors.
        Serial.println(&quot;&quot;);
        Serial.println(&quot;STOP&quot;);
        digitalWrite(Q1,LOW);
        digitalWrite(Q2,LOW);
        digitalWrite(Q3,LOW);
        digitalWrite(Q4,LOW);
}

void forward(){ // for move forward we need to on one side of the motor1 and same side of motor-2
        Serial.println(&quot;&quot;);
        Serial.println(&quot;Forward&quot;);
        digitalWrite(Q1,HIGH);
        digitalWrite(Q2,LOW);
        digitalWrite(Q3,HIGH);
        digitalWrite(Q4,LOW);
}

void backward(){
        Serial.println(&quot;&quot;);
        Serial.println(&quot;Backward&quot;);
        digitalWrite(Q1,LOW);
        digitalWrite(Q2,HIGH);
        digitalWrite(Q3,LOW);
        digitalWrite(Q4,HIGH);
}
```

```
void left(){
        Serial.println(&quot;&quot;);
        Serial.println(&quot;Left&quot;);
        digitalWrite(Q1,LOW);
        digitalWrite(Q2,HIGH);
        digitalWrite(Q3,HIGH);
        digitalWrite(Q4,LOW);
}

void right(){
        Serial.println(&quot;&quot;);
        Serial.println(&quot;Right&quot;);
        digitalWrite(Q1,HIGH);
        digitalWrite(Q2,LOW);
        digitalWrite(Q3,LOW);
        digitalWrite(Q4,HIGH);
}
```

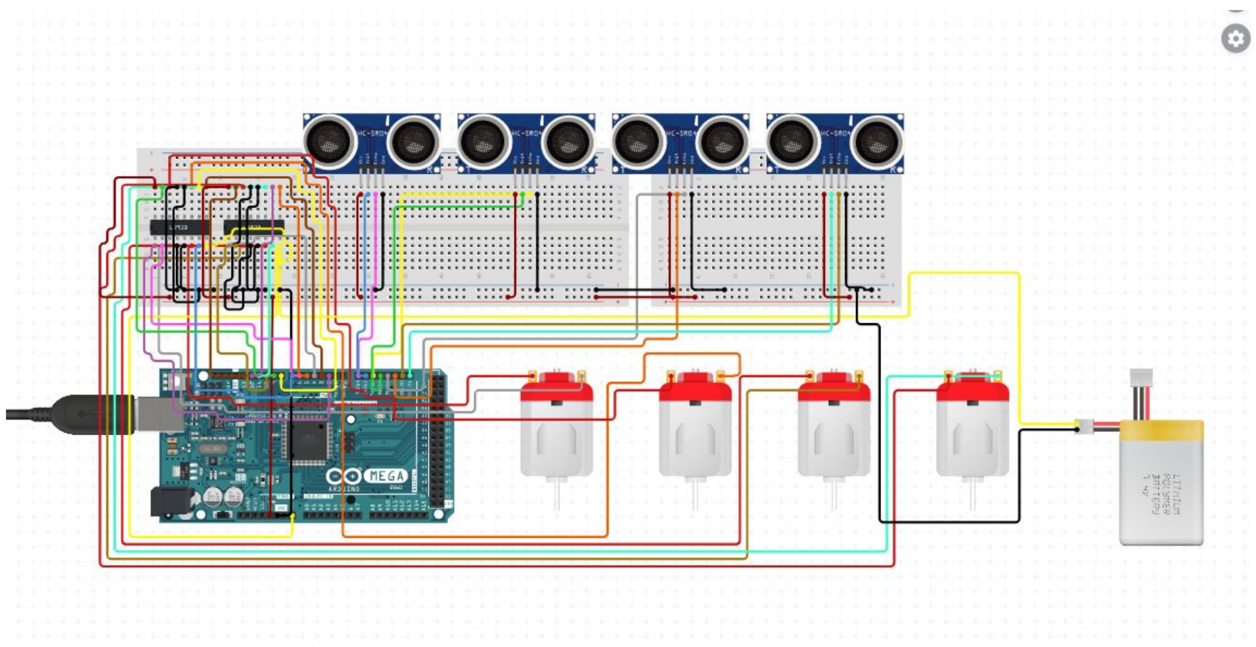
Final Parking spotter code

```
#define trigPin 13
#define echoPin 12

void setup() {
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
pinMode(5, OUTPUT); //pin of motor-A
pinMode(6, OUTPUT); //pin of motor-A
pinMode(9, OUTPUT); //pin of motor-B
pinMode(10, OUTPUT); //pin of motor-B
}

void loop() {
long duration, distance;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2); // wait for 2 seconds.
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1; //calculating the distance

if (distance < 7) {  // This is where the LED On/Off happens
    digitalWrite(6,HIGH);
    digitalWrite(5,LOW);
  }else { //we are performing the parking when the car finds at least 7 cm space.
    digitalWrite(9,LOW);
    digitalWrite(10,HIGH);
    analogWrite(6, 110);
    analogWrite(5, LOW);
    delay(50);

    digitalWrite(10,LOW);
    digitalWrite(9,HIGH);
    analogWrite(5, 110);
    analogWrite(6, LOW);
    delay(100);
```

```
   digitalWrite(9,LOW);
    digitalWrite(10,HIGH);
    analogWrite(5, 110);
    analogWrite(6, LOW);
    delay(100);

    digitalWrite(10,LOW);
    digitalWrite(9,LOW);
    analogWrite(5, LOW);
    analogWrite(6, LOW);
    delay(1000);
    }
}
```

Automatic car parking(extra feature)

Circuit:



Code:

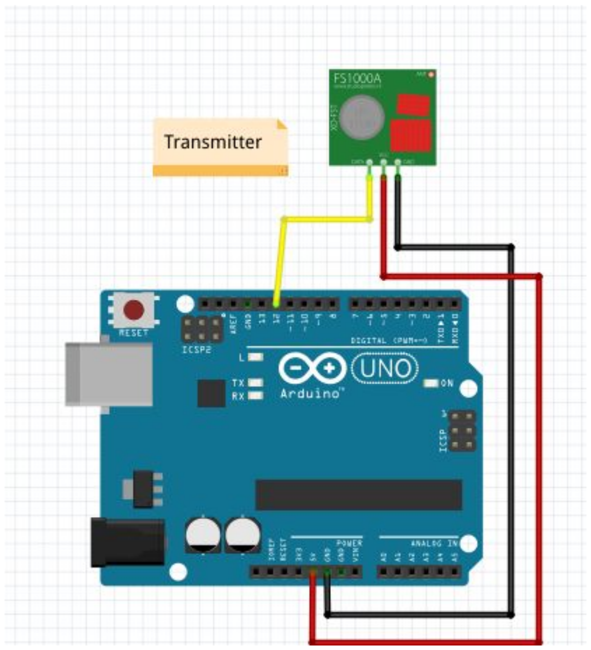
```
//defining for first ultrasonic
#define TRIGGER_PIN1 13
#define ECHO_PIN1 12
#define MAX_DISTANCE1 200
long dur1;
int dis1;
void Read () ;
//for second one
#define TRIGGER_PIN2 11
#define ECHO_PIN2 10
#define MAX_DISTANCE2 200
int dis2;
long dur2;

#define TRIGGER_PIN3 9
#define ECHO_PIN3 8
#define MAX_DISTANCE3 200
int dis3;
long dur3;
```

```
#define TRIGGER_PIN4 7
#define ECHO_PIN4 6
#define MAX_DISTANCE4 200
int dis4;
long dur4;

//defining the motors
int Q1LF = 2;
int Q2LR = 3;
int Q3RR = 4;
int in4RF = 5;

void setup() {
  pinMode(Q1LF, OUTPUT);
  pinMode(Q2LR, OUTPUT);
  pinMode(Q3RR, OUTPUT);
  pinMode(in4RF, OUTPUT);
  Serial.begin (9600);
}

void loop(){
  Read();
  if ((dis1 < 10 && dis3 < 15 && dis4 < 15) || (dis2 < 5 && dis3 < 15 && dis4 < 15)) { //we have
applied the left hand side approach in this case and finding the distance of all side base on that
we will decide to move in which direction
    digitalWrite (Q1LF, LOW);
    digitalWrite (Q2LR, LOW);
    digitalWrite (Q3RR, LOW);
    digitalWrite (in4RF, LOW);
  }else if (dis1 > 10) {
    digitalWrite (Q1LF, HIGH);
    digitalWrite (Q2LR, LOW);
    digitalWrite (Q3RR, LOW);
    digitalWrite (in4RF, HIGH);

if (dis3 < 55 && dis3 > 27) {
      digitalWrite(Q1LF, LOW);
      digitalWrite(Q3RR, LOW);
      digitalWrite(in4RF, LOW);
      digitalWrite(Q2LR, LOW);
      while (1) {
        digitalWrite(Q1LF, HIGH);
        digitalWrite(Q3RR, HIGH);
        digitalWrite(in4RF, LOW);
        digitalWrite(Q2LR, LOW);
        Read ();
        if (dis1 > 20 && dis3 > 40){
           break;
        }
      }
    }
    else if (dis4 < 55 && dis4 > 27) {
      digitalWrite(Q1LF, LOW);
      digitalWrite(Q3RR, LOW);
      digitalWrite(in4RF, LOW);
      digitalWrite(Q2LR, LOW);

      while (1) {
        digitalWrite(Q1LF, LOW);
        digitalWrite(Q3RR, LOW);
        digitalWrite(in4RF, HIGH);
        digitalWrite(Q2LR, HIGH);
        Read ();
        if (dis1 > 20 && dis4 > 40) {
           break;
        }
      }
    }
  }
  else if (dis1 < 10 && dis2 > 70 && dis3 < 15 && dis4 < 15){
    digitalWrite(Q1LF, LOW);
    digitalWrite(Q3RR, LOW);
    digitalWrite(in4RF, LOW);
    digitalWrite(Q2LR, LOW);

    while (1){
      digitalWrite(Q1LF, LOW);
      digitalWrite(Q3RR, HIGH);
      digitalWrite(in4RF, LOW);
      digitalWrite(Q2LR, HIGH);
```

```
      Read ();

      if (dis2 > 10) {
        break;
      }
    }
  }
}
void Read(){
  dis1= dur1*0.034/2; //calculating the distance
  Serial.print("Distance: "); // Prints the distance on the Serial Monitor
  Serial.println(dis1);
  dis2= dur2*0.034/2; //calculating the distance
  Serial.print("Distance: "); // Prints the distance on the Serial Monitor
  Serial.println(dis1);
  dis3= dur3*0.034/2; //calculating the distance
  Serial.print("Distance: "); // Prints the distance on the Serial Monitor
  Serial.println(dis1);
  dis4= dur4*0.034/2; //calculating the distance
  Serial.print("Distance: "); // Prints the distance on the Serial Monitor
  Serial.println(dis4);
}
```

RF Transmitter:

Circuit:

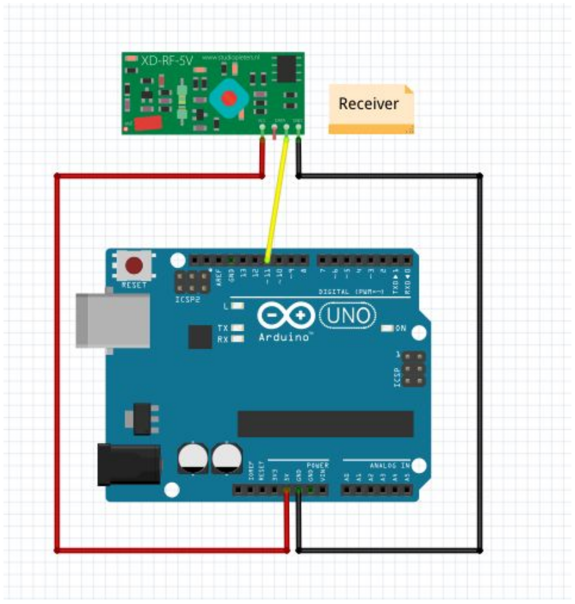


Code:

```
//TRANSMITTER
#include <RH_ASK.h>
#include <SPI.h>
RH_ASK driver;

void setup()
{
   Serial.begin(9600);
   if (!driver.init())
        Serial.println("init failed");
}

void loop()
{
   const char *msg = "Connected!";
   driver.send((uint8_t *)msg, strlen(msg));
   driver.waitPacketSent();
   delay(1000);
}
```

RF Receiver

Circuit:



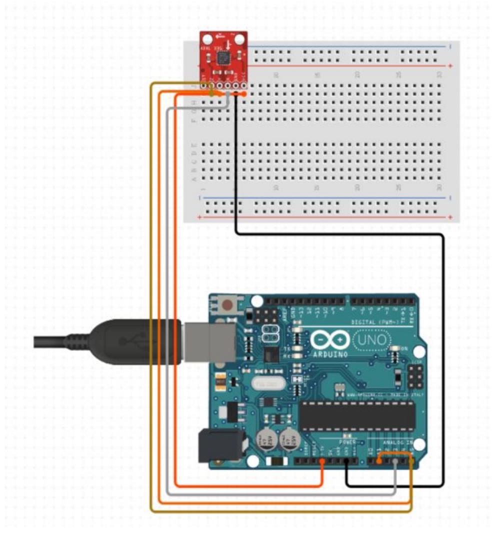
Code:

```
//RECEIVER
#include <RH_ASK.h>
#include <SPI.h>
RH_ASK driver;

void setup()
{
   Serial.begin(9600);
   if (!driver.init())
        Serial.println("init failed");
}
void loop()
{
   uint8_t buf[12];
   uint8_t buflen = sizeof(buf);
   if (driver.recv(buf, &buflen)){
     int i;
     Serial.print("Message: ");
     Serial.println((char*)buf);
   }
}
```

DESCRIPTION:

It receives the RF signal which helps our boat to use our command to move.

Accelerometer (ADXL355):
circuit:



Code:

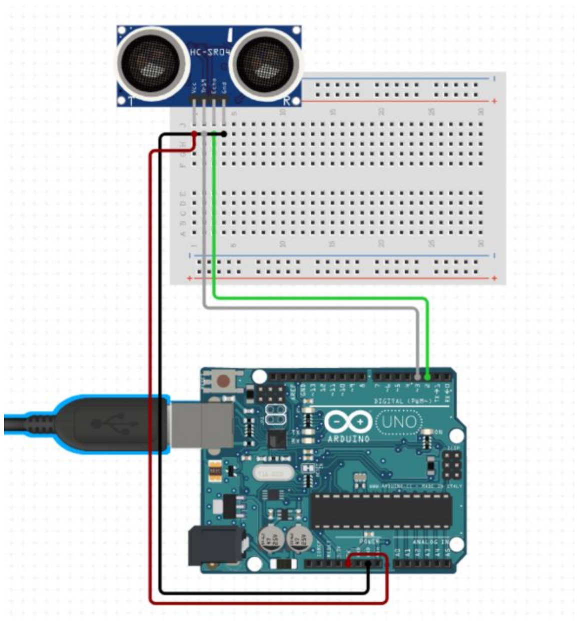
```
const int groundpin = 18; // analog input pin 4 -- ground
const int powerpin = 19; // analog input pin 5 -- voltage
const int xpin = A3; // x-axis of the accelerometer
const int ypin = A2; // y-axis of the accelerometer
const int zpin = A1; // z-axis of the accelerometer

void setup() {// initialize the serial communications:
  Serial.begin(9600);
  pinMode(groundpin, OUTPUT);
  pinMode(powerpin, OUTPUT);
  digitalWrite(groundpin, LOW);
  digitalWrite(powerpin, HIGH);
}
void loop() {
  Serial.print(analogRead(xpin)); // print the sensor values:
  Serial.print("\t"); // print a tab between values:
  Serial.print(analogRead(ypin));
  Serial.print("\t");// print a tab between values:
  Serial.print(analogRead(zpin));
  Serial.println();
  delay(100);// delay before next reading:
}
```

DESCRIPTION:

Here we used an accelerometer(ADXL355) which returned the value of the reading of the x,y,z axis.
Here it provides the value of the axis in the small change of the hand moment which leads to change in the value of any of the axis.

Ultrasonic sensor:

Circuit:



Code:

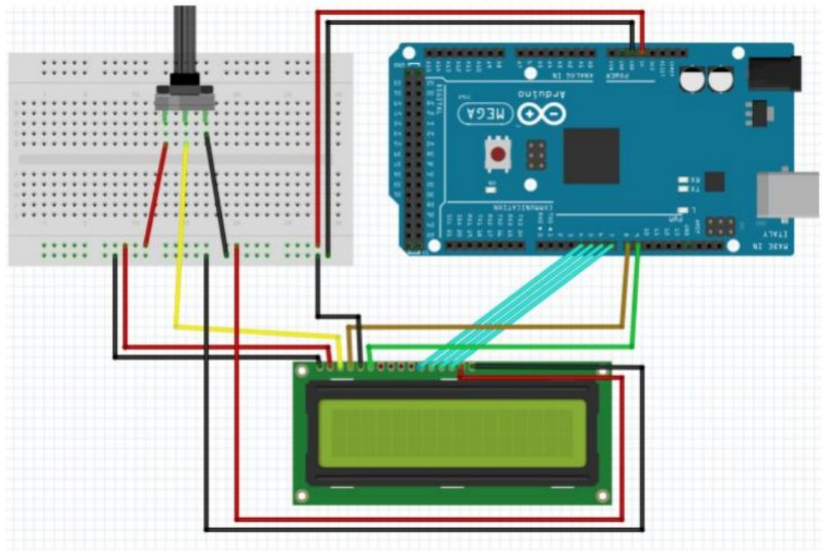
```
// defines pins numbers
const int trigPin = 9; //initializing the trigger pin
const int echoPin = 10;// initializing the echo pin of the ultrasonic sensor
long duration;
int distance;

void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
  pinMode(echoPin, INPUT); // Sets the echoPin as an Input
  Serial.begin(9600); // Starts the serial communication
}
void loop(){
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2); //wait for 2 microseconds.
  digitalWrite(trigPin, HIGH); // Sets the trigPin on HIGH state for 10 micro seconds
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin,HIGH);//Reads echoPin,returns travel time in microseconds
  distance= duration*0.034/2; //calculating the distance
  Serial.print("Distance: "); // Prints the distance on the Serial Monitor
  Serial.println(distance);
}
```

DESCRIPTION:

Here we used an ultrasonic sensor(HC-SR04) which returns the distance to the closest object in range. In order to do this, firstly it sends a pulse to the sensor to initiate a reading, then listens for a pulse to return.
And the length is calculated by this formula:
distance= duration*0.034/2

LCD Display:

Circuit Diagram



Code:

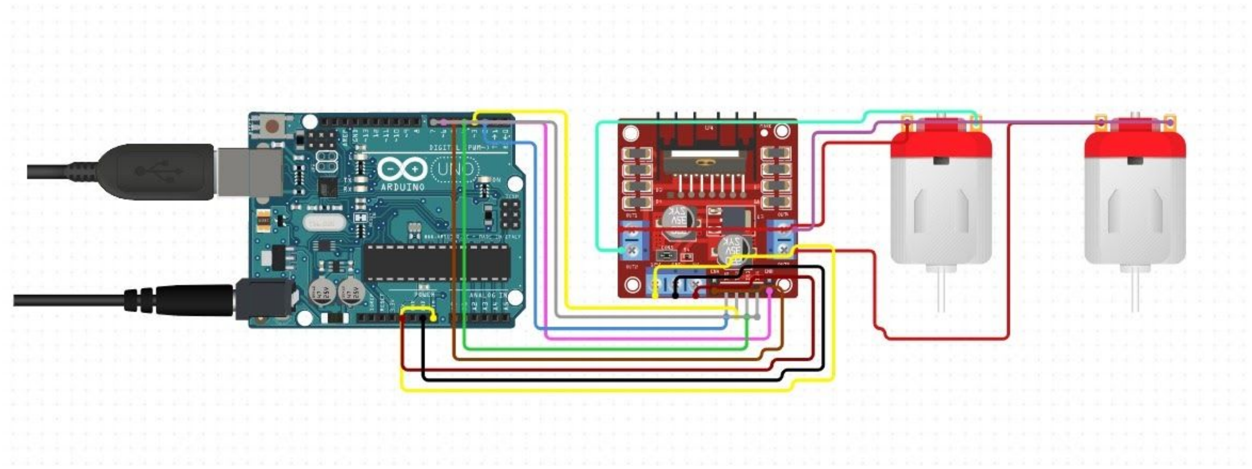
```
#include "LiquidCrystal.h"

// initialize the library by providing the number of pins to it
LiquidCrystal lcd(8,9,4,5,6,7);

void setup() {
lcd.begin(16,2);
// set cursor position to start of first line on the LCD
lcd.setCursor(0,0);
//text to print
lcd.print("Connected");
}

void loop(){
}
```

DESCRIPTION:

Here we used LCD (Liquid Crystal Display) which is a type of flat panel display which uses liquid crystals in its primary form of operation.

Motor driver(L293D) :

Circuit Diagram:



Code:

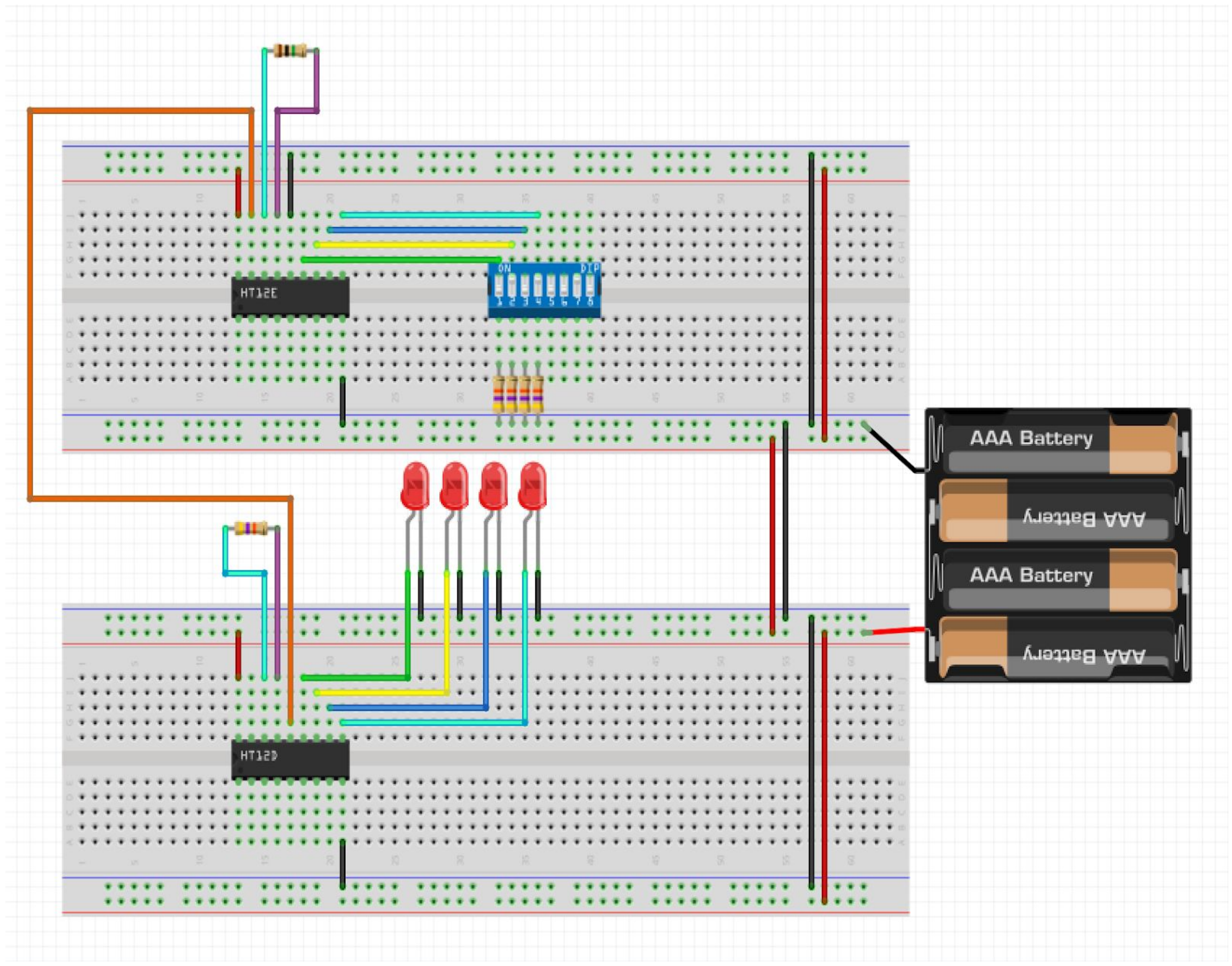
```
//L293D
//Motor A
const int motorPin1  = 9;  // Pin 14 of L293
const int motorPin2  = 10;  // Pin 10 of L293
//Motor B
const int motorPin3  = 6; // Pin  7 of L293
const int motorPin4  = 5;  // Pin  2 of L293
//This will run only one time.
void setup(){
   //Set pins as outputs
   pinMode(motorPin1, OUTPUT);
   pinMode(motorPin2, OUTPUT);
   pinMode(motorPin3, OUTPUT);
   pinMode(motorPin4, OUTPUT);
   //Motor Control - Motor A: motorPin1,motorpin2 & Motor B: motorpin3,motorpin4

   //This code  will turn Motor A clockwise for 2 sec.
   analogWrite(motorPin1, 180);
   analogWrite(motorPin2, 0);
   analogWrite(motorPin3, 180);
   analogWrite(motorPin4, 0);
   delay(5000);
   //This code will turn Motor A counter-clockwise for 2 sec.
   analogWrite(motorPin1, 0);
```

```
  analogWrite(motorPin2, 180);
  analogWrite(motorPin3, 0);
  analogWrite(motorPin4, 180);
  delay(5000);

  //This code will turn Motor B clockwise for 2 sec.
  analogWrite(motorPin1, 0);
  analogWrite(motorPin2, 180);
  analogWrite(motorPin3, 180);
  analogWrite(motorPin4, 0);
  delay(1000);
  //This code will turn Motor B counter-clockwise for 2 sec.
  analogWrite(motorPin1, 180);
  analogWrite(motorPin2, 0);
  analogWrite(motorPin3, 0);
  analogWrite(motorPin4, 180);
  delay(1000);

  //And this code will stop motors
  analogWrite(motorPin1, 0);
  analogWrite(motorPin2, 0);
  analogWrite(motorPin3, 0);
  analogWrite(motorPin4, 0);
}
```

DESCRIPTION:

Here we used the L293D motor driver IC which is basically a dual H-Bridge motor driver that allows speed and direction control of two DC motors at the same time.

HT-12D and HT-12E(encoder and decoder IC):



DESCRIPTION:

This IC plays an important role in encoding and decoding the message which is sent by RF transmitter and Receiver.

REFERENCE:

- https://create.arduino.cc/projecthub
- **https://www.instructables.com/id/Gesture-controlled-robot-using-Arduino/**
- **https://create.arduino.cc/projecthub/mayooghgirish/hand-gesture-controlled-robot-4d7587**
- **https://howtomechatronics.com/tutorials/arduino/**
- **https://www.allaboutcircuits.com/projects/interface-an-lcd-with-an-arduino/**
- **http://tok.hakynda.com/article/detail/177/ht12d-ht12e-communicate-through-wire**