MAKERSPACE-2K19

GESTURE CONTROLLED ROBOT

AIM: TO CONTROL THE ROBOT THROUGH GESTURES VIA A RF MODULE

GROUP MEMBERS:

- YUKTA.P.SHAH
- MANASVI.SINHA
- SAHIL NARE
- YASHWANTH REDDY
- JAY VAGHASIYA
- DHWAJ KOTHARI

GROUP MENTORS:

- DHIMAN AIRAO
- VYBHAV NEELISETTY
- MOHIT GIDWANI

COMPONENTS USED:

1. ATMEGA 128(Microcontroller)
2. MPU 6050
3. L239D(Motor driver)
4. RX433 AND TX433 (RF MODULE)
5. USB ASP PROGRAMMER
6. TRANSISTOR TRANSISTOR LOGIC (TTL)
7. JUMPERS
8. VEGA MOTORS (12V ,300 RPM)
9. 7805

SOFTWARE USED:

- ATMEL STUDIO 7.0
- Extreme burner AVR
- XCTU

COMMUNICATION PROTOCOL USED:

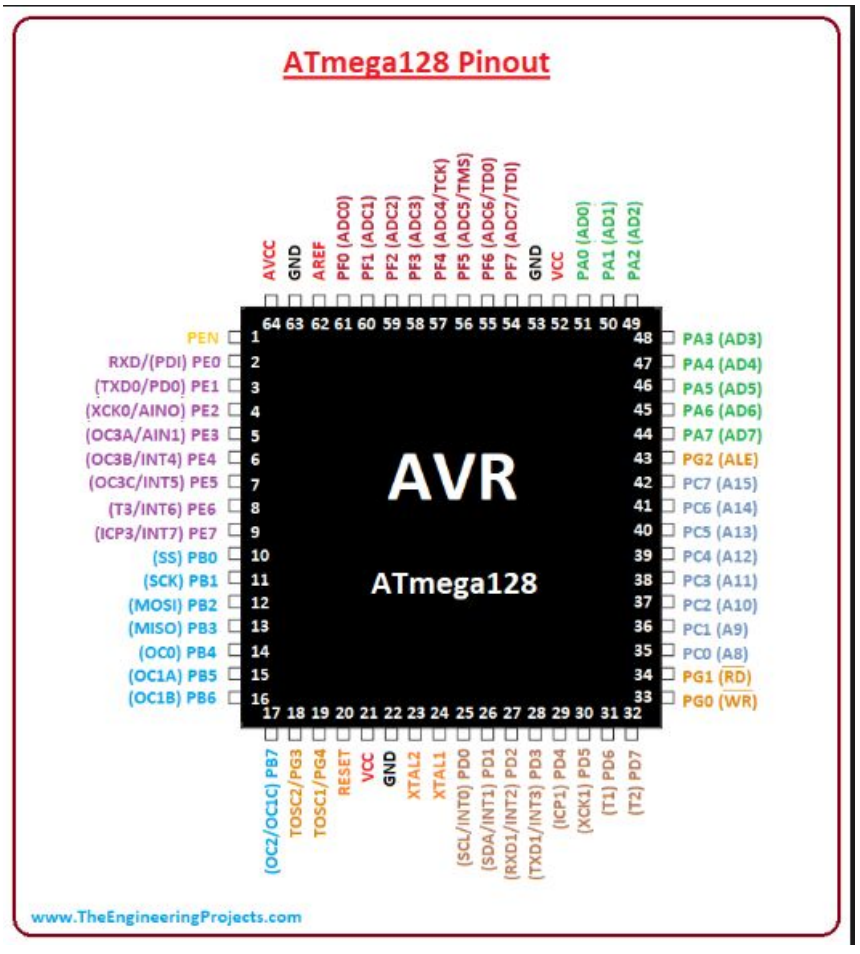
- UART COMMUNICATION
- I2C COMMUNICATION

BRIEF DESCRIPTION OF COMPONENTS:

1. ATMEGA 128:

   The high-performance, low-power Microchip 8-bit AVR RISC-based microcontroller combines 128KB of programmable flash memory, 4KB SRAM, a 4KB EEPROM, an 8-channel 10-bit A/D converter, and a JTAG interface for on-chip debugging. The device supports throughput of 16 MIPS at 16 MHz and operates between 4.5-5.5 volts.

   By executing instructions in a single clock cycle, the device achieves throughputs approaching 1 MIPS per MHz, balancing power consumption and processing speed.

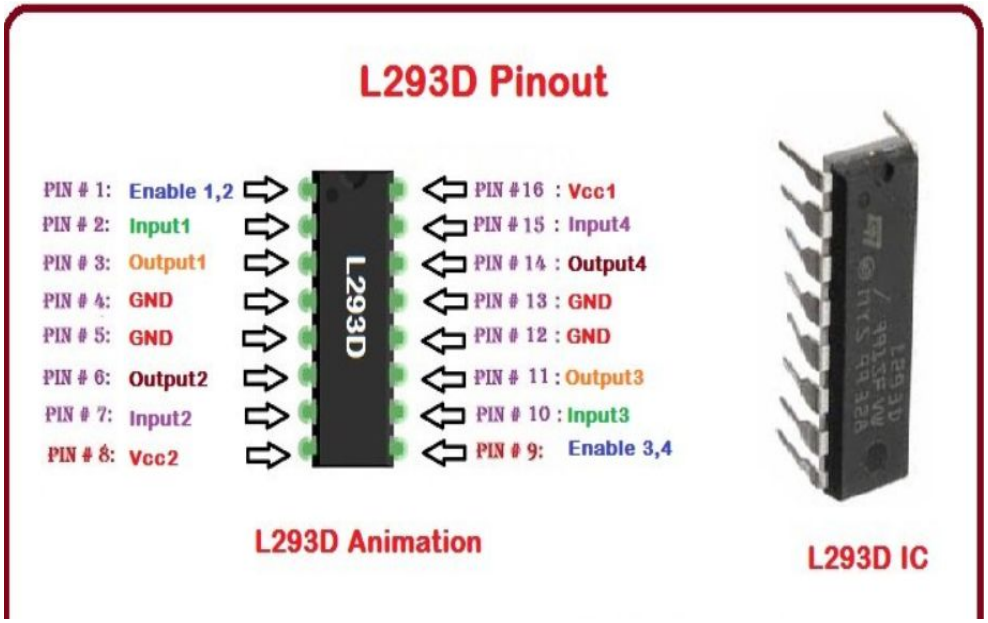


PARAMETERS:

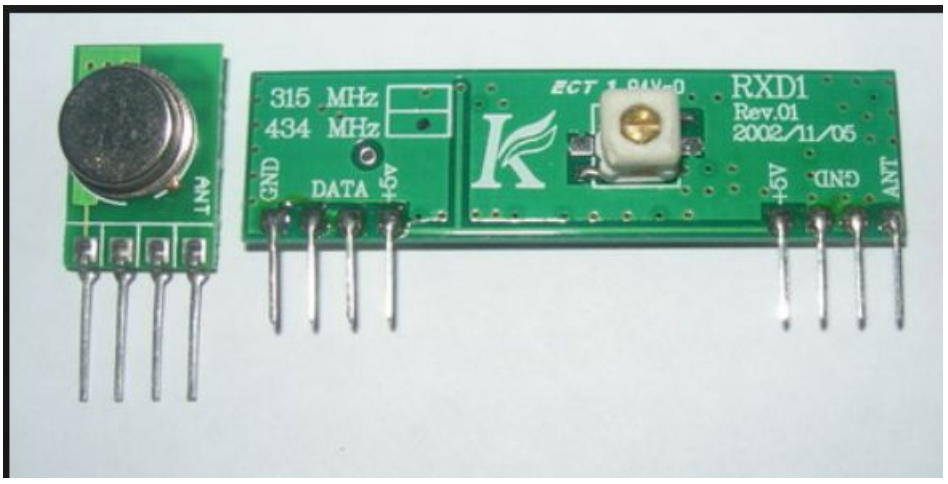
Name	Value
Program Memory Type	Flash
Program Memory Size (KB)	128
CPU Speed (MIPS/DMIPS)	16
SRAM Bytes	4,096
Data EEPROM/HEF (bytes)	4096
Digital Communication Periphe...	2-UART, 1-SPI, 1-I2C
Capture/Compare/PWM Periph...	2 Input Capture, 2 CCP, 8PWM
Timers	2 x 8-bit, 2 x 16-bit
Number of Comparators	1
Temperature Range (C)	-40 to 85
Operating Voltage Range (V)	2.7 to 5.5
Pin Count	64

2. MPU 6050:
   - ➢ It is a sensor which consists of an accelerometer, a gyroscope, and temperature sensor. It gives raw values (values without any manipulation) and we need to convert the data to logical data using mathematical operations in atmega.
   - ➢ It utilises the I2C communication protocol.

3. L293D:
   - ➢ It is a motor driver which consists of 4 inputs and 4 outputs.
   - ➢ 2 inputs and 2 outputs for each motor.
   - ➢ It drives motors using signals of a microcontroller (atmega in our case)



4. RX433 AND TX433 (RF MODULE):
   - ➢ These are Radio Frequency receiver and transmitter respectively.
   - ➢ They are used to transmit data wirelessly using radio frequency.



5. USB ASP PROGRAMMER:

➢ Used to burn codes in AVR programmers

(To burn code means transfer the source code to the atmega)

5 TRANSISTOR TRANSISTOR LOGIC (TTL):

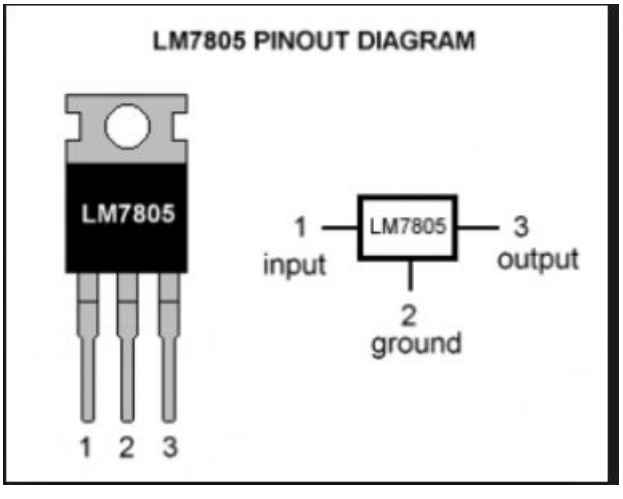
➢ Used to check communication between two AVR controllers.

➢ They communicate using USART or UART protocols.

6 7805 :

➢ Works as a voltage regulator.

➢ It converts any voltage supply above 5volts to 5volts.



VEGA MOTORS:

This are used for driving the motors.it has a 12 volt supply with 300rpm.

The communication protocol:

I2C COMMUNICATION :

This is the communication in gesture controlled robot for the interface between the MPU6050 and ATMEGA. The mpu6050 communicates only through I2C communication whereas the ATMEGA can communicate through I2C, UART AND SPI as well.

The Two-wire Serial Interface (TWI) is ideally suited for typical microcontroller applications. The
TWI protocol allows the systems designer to interconnect up to 128 different devices using only
two bi-directional bus lines, one for clock (SCL) and one for data (SDA). The only external hardware
needed to implement the bus is a single pull-up resistor for each of the TWI bus lines. All devices connected to the bus have individual addresses, and mechanisms for resolving bus
contention are inherent in the TWI protocol.

The procedure of TWI communication

1. The first step in a TWI transmission is to transmit a START condition. This is done by writing a specific value into TWCR, instructing the TWI hardware to transmit a START condition. Which value to write is described later on. However, it is important that the TWINT bit is set in the value written. Writing a one to TWINT clears the flag. The TWI will not start any operation as long as the TWINT bit in TWCR is set. Immediately after the application has cleared TWINT, the TWI will initiate transmission of the START condition.

2. When the START condition has been transmitted, the TWINT flag in TWCR is set, and TWSR is updated with a status code indicating that the START condition has successfully
been sent.

3. The application software should now examine the value of TWSR, to make sure that the START condition was successfully transmitted. If TWSR indicates otherwise, the application software might take some special action, like calling an error routine. Assuming that the status code is as expected, the application must load SLA+W into TWDR. Remember that TWDR is used both for address and data. After TWDR has been loaded with the desired SLA+W, a specific value must be written to TWCR, instructing the TWI hardware to transmit the SLA+W present in TWDR. Which value to write is described later on. However, it is important that the TWINT bit is set in the value written. Writing a one to TWINT clears the flag. The TWI will not start any operation as long as the TWINT bit in TWCR is set. Immediately after the application has cleared TWINT, the TWI will initiate transmission of the address packet.

4. When the address packet has been transmitted, the TWINT flag in TWCR is set, and TWSR is updated with a status code indicating that the address packet has successfully been sent. The status code will also reflect whether a slave acknowledged the packet or not.

5. The application software should now examine the value of TWSR, to make sure that the address packet was successfully transmitted, and that the value of the ACK bit was as expected. If TWSR indicates otherwise, the application software might take some special action, like calling an error routine. Assuming that the status code is as expected, the application must load a data packet into TWDR. Subsequently, a specific value must be written to TWCR, instructing the TWI hardware to transmit the data packet present in TWDR. Which value to write is described later on. However, it is important that the TWINT bit is set in the value written. Writing a one to TWINT clears the flag. The TWI will not start any operation as long as the TWINT bit in TWCR is set. Immediately after the application has cleared TWINT, the TWI will initiate transmission of the data packet.

6. When the data packet has been transmitted, the TWINT flag in TWCR is set, and TWSR is updated with a status code indicating that the data packet has successfully been sent. The status code will also reflect whether a slave acknowledged the packet or not.

7. The application software should now examine the value of TWSR, to make sure that the data packet was successfully transmitted, and that the value of the ACK bit was as expected. If TWSR indicates otherwise, the application software might take some special action, like calling an error routine. Assuming that the status code is as expected, the application must write a specific value to TWCR, instructing the TWI hardware to transmit a STOP condition. Which value to write is described later on. However, it is important that the TWINT bit is set in the value written. Writing a one to TWINT clears the flag. The TWI will not start any operation as long as the TWINT bit in TWCR is set. Immediately after the application has cleared TWINT, the TWI will initiate transmission of the STOP condition.
Note that TWINT is NOT set after a STOP condition has been sent.

TWI Control Register – TWCR

Bit	7	6	5	4	3	2	1	0	
	TWINT	TWEA	TWSTA	TWSTO	TWWC	TWEN	–	TWIE	TWCR
Read/Write	R/W	R/W	R/W	R/W	R	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

USART COMMUNICATION PROTOCOL:

The Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART) is a
highly flexible serial communication device. The main features are:
• Full Duplex Operation (Independent Serial Receive and Transmit Registers)
• Asynchronous or Synchronous Operation
• Master or Slave Clocked Synchronous Operation
• High Resolution Baud Rate Generator
• Supports Serial Frames with 5, 6, 7, 8, or 9 Data Bits and 1 or 2 Stop Bits
• Odd or Even Parity Generation and Parity Check Supported by Hardware
• Data OverRun Detection
• Framing Error Detection
• Noise Filtering Includes False Start Bit Detection and Digital Low Pass Filter
• Three Separate Interrupts on TX Complete, TX Data Register Empty, and RX Complete
• Multi-processor Communication Mode
• Double Speed Asynchronous Communication Mode

The ATmega128 has two USART's, USART0 and USART1. The functionality for both USART's
is described below. USART0 and USART1 have different I/O registers.

PRIMARY APPROACH :

- We decided the components and the basic layout of our gesture controlled robot .
  The basic layout was : on the glove:
  Sensor(mpu6050) ---atmega32----encoder-----RF transmitter
  On the robot:

RF receiver-----decoder-------atmega32--------L293D(motor driver)
We understood the form of input and output in each case like for the sensor the input is analog and the output is also analogue signal.
I2C communication is used between sensor and the atmega32.
The UART communication is to be used between atmega32 and atmega32.

MODIFIED APPROACH:

- With a better understanding of the UART communication and how the encoder decoder works, a decision was taken to not use encoder and decoder as the same task was performed by the UART communication between the two atmegas.
  Due to problems encountered with the atmega32, an alternative ,that is, atmega128 is used.
  The group was divided to work on the I2C code , UART code and circuit designing.
  On the completion of the code and after succeeding in obtaining the raw values from the mpu6050 , complimentary filters (combination of low pass filter and high pass filter for the accelerometer and the gyroscope respectively) needs to be implemented in the code to reduce the variations in the values due to the interference.

REFERENCE LINKS:

- Our code link on github:
https://github.com/sahilnare/gesture_controlled_robot
- http://www.electronicwings.com/images/topic_content_attachments/Thu-06-16-22-56-51.I2C_Master_C_file.c
- https://www.avrfreaks.net/forum/i2c-readingwriting-problems
- https://www.youtube.com/watch?v=JMMamSVy1Zs&list=PLE72E4CFE73BD1DE1
- http://www.pieter-jan.com/node/11
- https://www.youtube.com/watch?v=qmd6CVrlHOM