Hello everyone, welcome to another video on proteus simulation by Robotics & Circuits, Manipal. We sincerely hope you enjoyed our previous videos. Today, we will build a Bluetooth-controlled Robotic Arm using Arduino Mega 2560.

We have provided you with a link to install Arduino Mega 2560 in Proteus in the description below.

<https://www.theengineeringprojects.com/2015/12/arduino-mega-2560-library-proteus.html>

Arduino Mega 2560 of the Arduino family is specially designed for projects that require more I/O pins, more sketch memory, and RAM. It is a microcontroller board based on the ATmega2560. The board consists of 54 I/O pins, of which fifteen pins can be used as PWM pins. It also consists of 4 UARTS (hardware serial ports), 16 analog input pins, 16 MHz crystal oscillator, a power jack, an ICSP header, a USB connection, and a reset button. It comes with a flash memory of 256 kB, which is 8x more memory space than Arduino Uno and Mini, allowing us to sketch more extensive codes. The Mega 2560 board has SRAM space of 8kB (which is again 4x more than Uno), allowing Arduino to create and manipulate more variables when the program runs. The board also consists of 4kB EEPROM memory, 3 Rx and Tx pins, 1 SDA and SCL pins. The Mega 2560 board is compatible with nearly all shields designed for Uno.

Now that we have familiarised ourselves with Mega 2560, let us now start making our Robotic Arm. We will control the Arm via Bluetooth.

In this video, we will be using Servo motors, allowing us precise control of arm movements. A servo motor is a rotary/linear actuator that allows accurate control of angular positions. The motor is coupled with a feedback sensor for position feedback. It also consists of a servo drive that uses the feedback sensor to precisely control the rotary position. The motors are controlled by sending PWM signals from the controller. The PWM sent to the motor determines the initial position of the shaft of the motor, and based on the PWM signal, it will turn to the desired position.

The motor consists of three pins-

1. Vcc

2. GND

3. The signal wire (PWM signal is sent to this pin)

We will be using four servo motors as base, shoulder, elbow, and Gripper of the robotic Arm.

1. The base of the Arm will be turned using a servo motor.

2. The shoulder will be responsible for lowering/raising the Arm vertically.

3. The elbow makes the Arm move forward/backward.

4. The Gripper will be used to grab the things.

To communicate with the Arm, we will be using the HC-06 Bluetooth module. The communication with the module is done through the UART interface. The reason for selecting this module instead of HC-05 is that this particular module is specially built for short-range wireless data communication. It works on Bluetooth 2.0 communication protocol and can only act as a slave. Hc-06 uses the FHSS technique (frequency hopping spread spectrum) to avoid interference with other devices and to have Full duplex transmission. FHSS is a method in which signals are transferred by rapidly changing carrier frequency.

We have also provided you with the link for the HC-06 module in the description below.

<https://www.theengineeringprojects.com/2016/03/bluetooth-library-for-proteus.html>

Like the last video, we will be using an alphanumeric 16x2 LCD to display the actions performed by the Arm. We have modified the 16x2 LCD available in the Proteus to make it 16 pins LCD instead of 14 pins. You can watch the tutorial below to adjust the LCD in Proteus.

<https://www.youtube.com/watch?v=bsmObOS2Ov8>

Now, we can also control the Robotic Arm in manual mode instead of maneuvering it via Bluetooth. For this purpose, we will be using potentiometers to give analog input to the Mega2560. We have also attached three switches for program, run, and reset, respectively.

Now that the construction of the Robotic Arm is over, we can add the code to Arduino Mega2560.

**The Code:**

Though not complicated, the code contains many variables.

In ArmDefine.h, we have set the minimum, maximum and initial servo angles for each position.

Some of the notable functions defined are **checkBTcmd()**, **defineLocalRemote()**, **recArmPosition()**. The **checkBTcmd()** assembles a string using the information arriving from the Bluetooth module. The **defineLocalRemote()** toggles the control mode between manual and wireless depending on the input. On the other hand, **recArmPosition()** writes the step position for the servo motors.

We have also provided the Bluetooth app apk in a separate folder.

And this is how the Bluetooth Controlled Robotic Arm works!!!

[paste the microcontroller code here]

By

* Abhishek Srinivas Rao
* Yadav Ratnesh Ravikiran