**CHAPTER 4**

**DESIGN CONSIDERATION OF ROBOTIC ARM CONTROL SYSTEM**

In this chapter, program flow chart and circuit operation of the system are explained in details.

**4.1. Design Concept of the System**

In this process, the system is built with Android phones, Bluetooth module (HC-06), Arduino Mega, motor driver (L298), CA 2596 step down (DC-DC) converter, DC gear motors and servo motors. Android phones are used for Bluetooth. Bluetooth module (HC 06) receives the data from android phones and sends the data to the Arduino Mega. Arduino Mega is used as the heart of the entire system. Motor driver (L298) is used for drive the DC gear motors. Servo motor is used to control the gripper.

**4.2. Overall Circuit Operation of the System**

The Arduino Mega is the main part of the system. Overall circuit diagram of the system is shown in Figure 4.1.



Figure 4.1. Overall Circuit Diagram of the System

Firstly, for power supply, 14.8V battery is used for Arduino Mega, Motor driver(L298) and CA 2596 Step Down (DC-DC) converter. 5V in Arduino Mega is connected to Bluetooth Module (HC 06) and 5V in CA 2596 Step Down (DC-DC) converter is also connected to four Servo motors. Bluetooth module of pin Tx and Rx connect D0 and D1 of the Arduino Mega. Arduino Mega D2, D3, D4, D5 pins connect IN1, IN2, IN3, IN4 of the Motor driver L298. Servo motor1, Servo motor2, Servo motor3 and Servo motor4 of pins are connected to D8, D9, D10, D11 of the Arduino Mega. DC motor 1 and motor 4 are connected to the OUT1 and OUT2 of the Motor driver L298. DC motor 2and motor 3 are connected to the OUT3 and OUT4 of the Motor driver L298.

**4.3. Overall Flow Chart Operation of the System**

The Arduino is programed by using the C++ language using Arduino IDE. At the first of the program, power on the robot and open user phone Bluetooth and BTP pick Vehicle android application. And then it pairs with Bluetooth module of robot. If there is connection, android application is ready to connect with robot. If not, going back to the pairing with Bluetooth module of robot. And then android application is ready to send input command to the robot. Read keys data from android phone, check the data > 0 and then all motors on. If not, all motors off.

If data=1, the phone-controlled robot moving to forward and if not, give next input command. If data=2, the phone-controlled robot moving to backward and if not this direction, give the next input command. If data=3, the phone-controlled robot moving to left. If not this direction, give the next input direction. If data=4, the phone-controlled robot moving to right direction. If not, going back to the input command.

If data=8 and 9, turn servo motor M4 on (clock and counter clockwise) to make swing (waist in app) left and right and if not, given next input command. If data=10 and 11, turn servo motor M3 on (clockwise and counter clockwise) to make swing (shoulder in app) forward and backward and if not, given next input command. If data=12 and 13, turn servo motor M2 on (clock and counter clockwise) to make swing (elbow in app) upward and downward and if not given next input command. If data=14 and 15, turn servo motor M1 on (clock and counter clockwise) to make gripper open and close functions. These processes are repeated the end of the process of pick and place. The control of the program can easily be implemented. Figure 4.2 shows the overall flow chart of robotic arm control system.



Figure 4.2. Overall Flow Chart of the System

**4.4. Creating the Application for Robotic Arm Control System**

In this project, the user developed an application using Arduino and Bluetooth communication. The application was created using MIT App Inventor, a web-based platform that allows for easy development of Android applications. The UI of the application of the robotic arm control system is shown in Figure 4.3.



Figure 4.3. UI of the Application of the Robotic Arm Control System

4.4.1. Accessing MIT App Inventor Website

Firstly, the MIT App Inventor website is accessed. Then, the user encounters two main sections: “designer” and “blocks”. Figure 4.4 shows the MIT App inventor website.

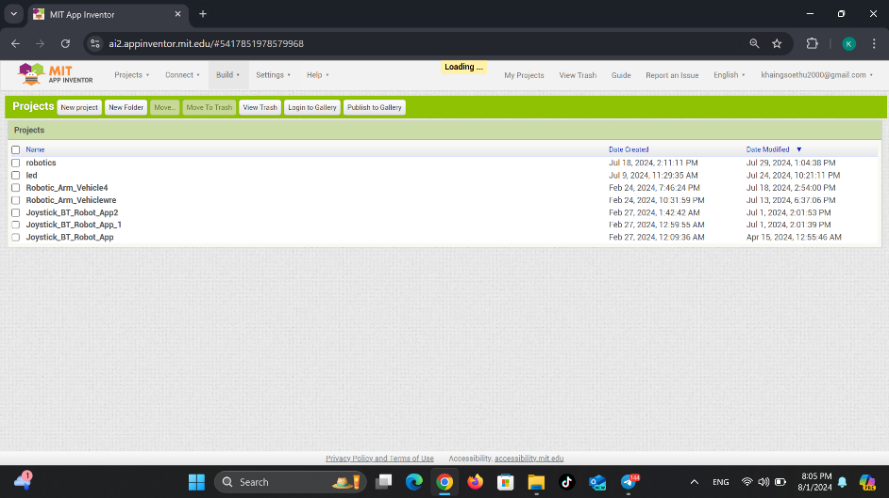


Figure 4.4. MIT App Inventor Website

4.4.2. Designing the User Interface

By using the “designer” section, the user interface is crested for the application. This section allows us to drag and drop various components to design the layout of the app. The design is able to be adjusted according to the size of the screen size of the device. For the project, buttons for controlling the vehicle, the robotic arm's movement and gripper functions are included. The user interface was designed to be intuitive with clearly buttons for moving forward, backward, left, and right. Figure 4.5 shows the “Designer” section of MIT App inventor.

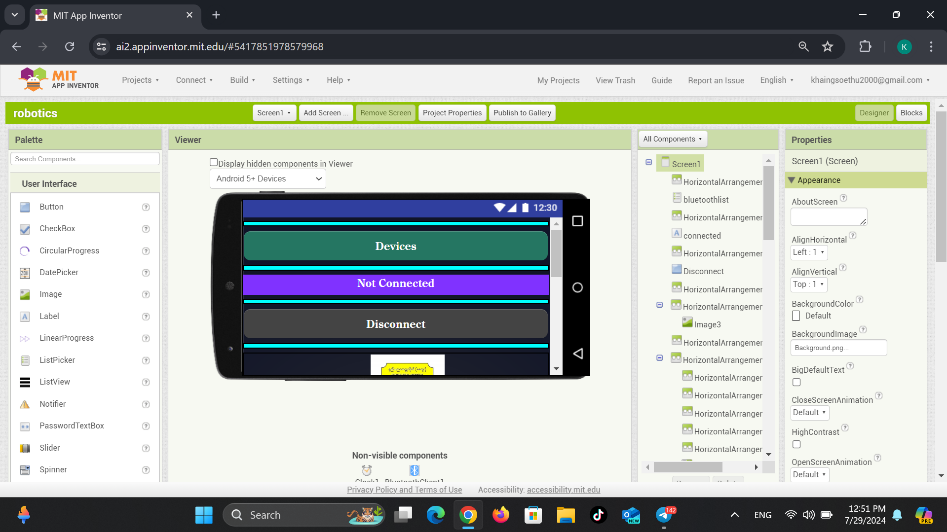


Figure 4.5. Designer Section of MIT App Inventor

4.4.3. Programming the Functions for the App

By using the “blocks” section, the functionality of each button is defined. The Blocks section provides a visual programming environment where different blocks to create instructions are able to be connected. For the application, user programmed the buttons to send specific Bluetooth signals corresponding to the commands: front, back, left, right, and gripper control. These signals are then interpreted by the Arduino to control the robotic arm. Data 1, 2, 3 and 4 are used for moving vehicle forward, backward, left and right. Data 5 is used to stop the vehicle. Data 8 and 9 are used for the griper to open and close, Data 10 and 11 are used for upward and downward movement of elbow. Data 12 and 13 are used for upward and downward movement of the shoulder. Data 14 and 15 are used for left and right movement of the waist or base.

Figure 4.6 shows the “Block” section of MIT App inventor.

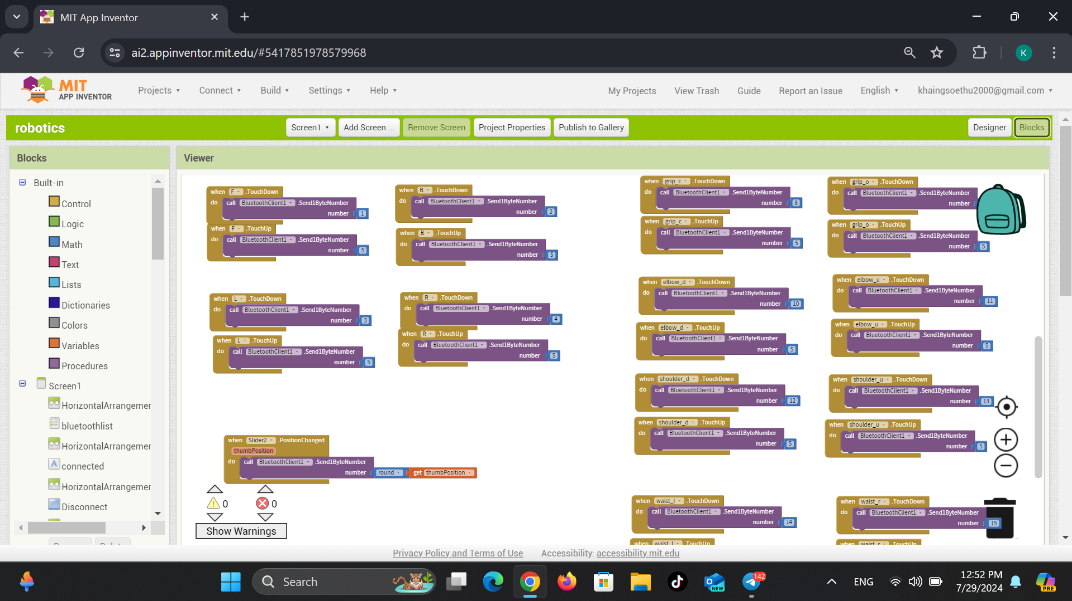


Figure 4.6. Block section of MIT App Inventor

4.4.4. Building Bluetooth Application

After completing the user interface design and programming the functionality, the user proceeded to build the application. Click on the Build button located beside the Settings option. This action opens a menu with two options: Android App (.apk) and Android App Bundle (.aab). Select Android App (.apk) to compile the application. Figure 4.7 shows building Bluetooth application.

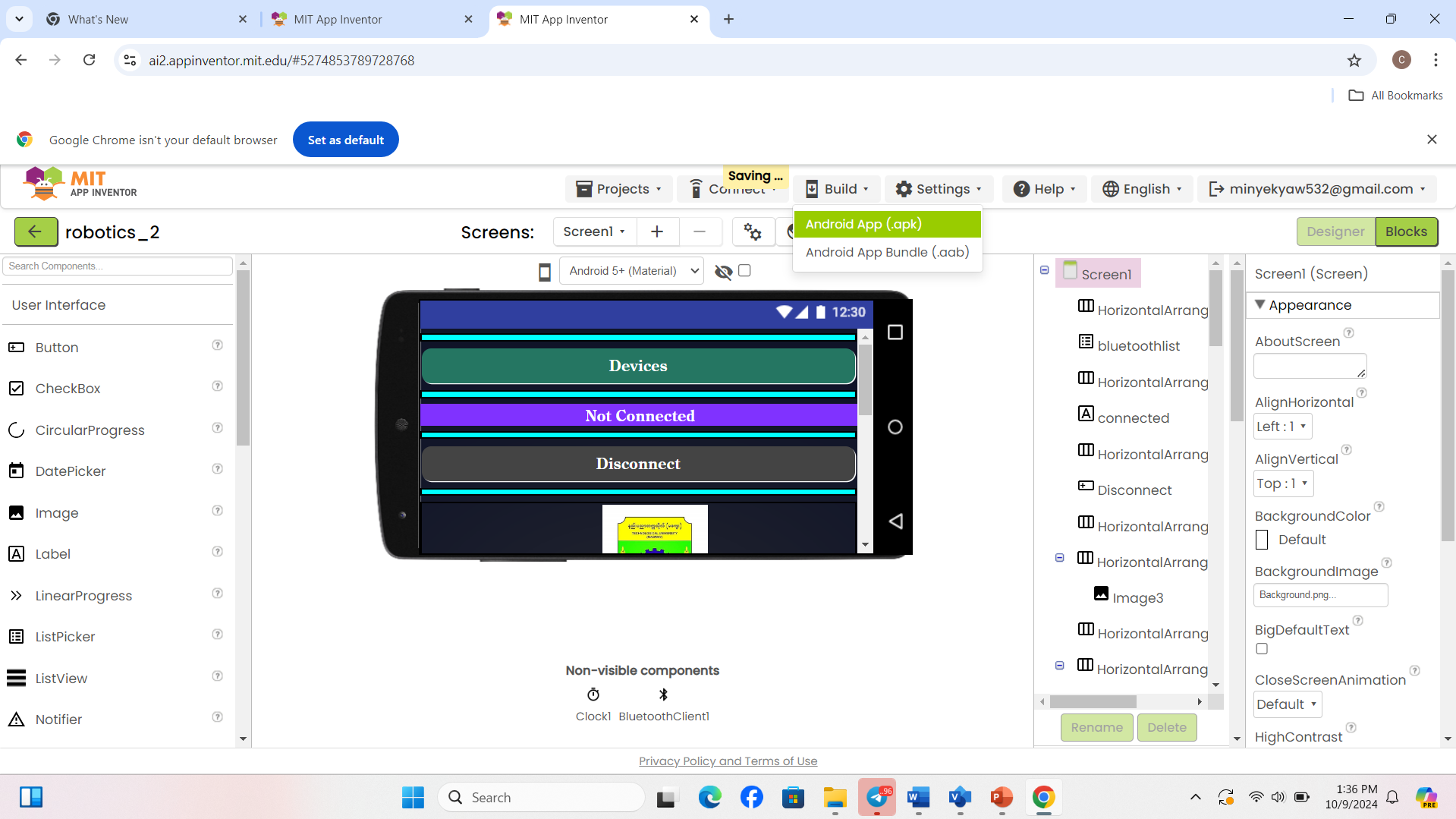


Figure 4.7. Building Bluetooth Application

4.4.5. Compiling and Downloading Bluetooth Application

In this section, there are two options: the android app (.apk) and the android app bundle (.aab). After choosing the Android App(.apk), the compilation process will begin and progress will be displayed on the screen. When the process reaches 100%, a download link and a barcode scanner code for the application will be provided. The user can use the download link to directly download the app or scan the barcode to install it on an Android device. The installed app in the android phone is ready to use for controlling the robotic arm. Figure 4.8 shows compiling Bluetooth application. Figure 4.9 shows downloading Bluetooth application.

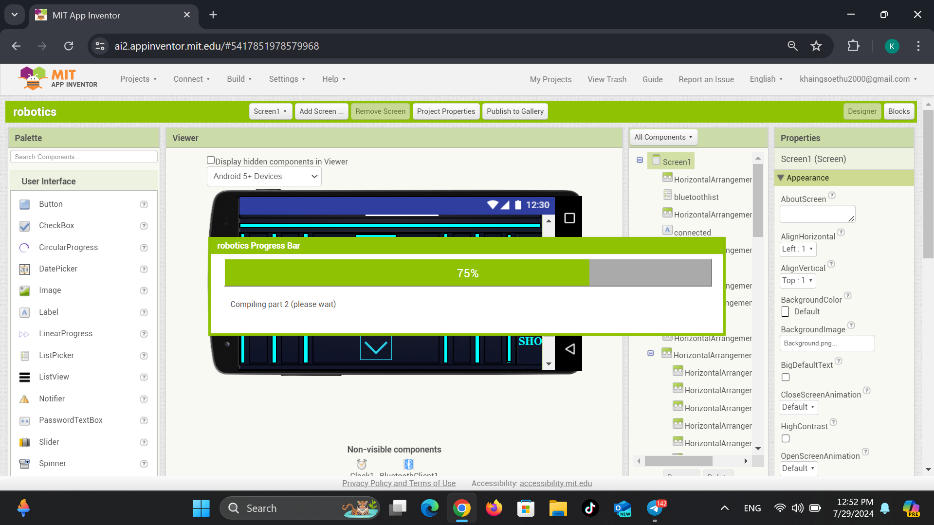


Figure 4.8. Compiling Bluetooth Application

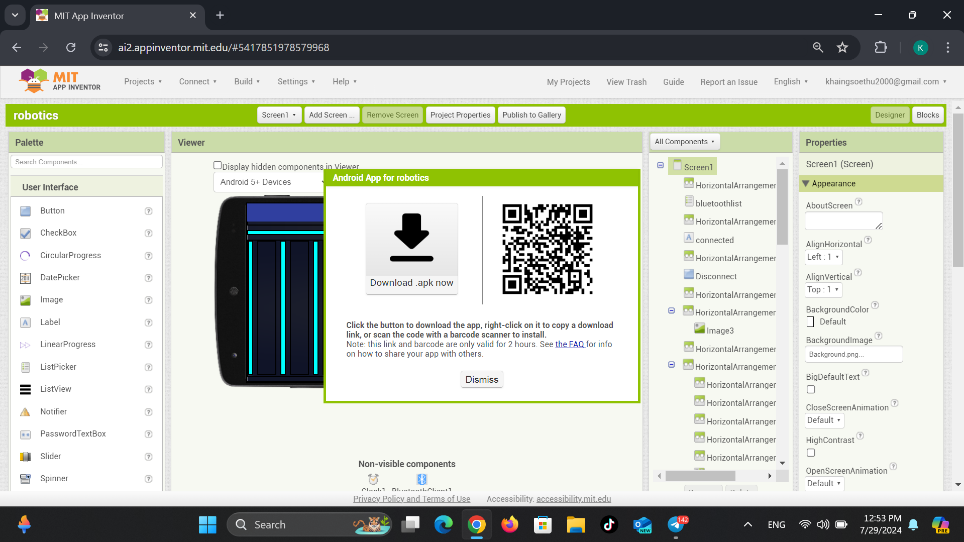


Figure 4.9. Downloading Bluetooth Application.

**4.5. Summary**

This chapter has been discussed about the flow chart, design concept and circuit diagram operation for this project. The next chapter will be described about the tests and results of the robotic arm control system.