**ภาคผนวก ค**

โปรแกรมอะดุยโน ไอดีอี บอร์ดอีเอสพีโน 32

โปรแกรมอะดุยโน ไอดีอี บอร์ดอีเอสพีโน 32

|  |  |
| --- | --- |
| 1 | //----------------------------------------------------------------------------- |
| 2 | // Project : RMUTR Digital Lock V1 |
| 3 | // VSCode Extension : PlatformIO IDE 1.10.0 |
| 4 | // Source : https://github.com/rmutr/Digital\_Lock\_V1\_Arduino.git |
| 5 | // Board : Node32s (Gravitech Node32Lite LamLoei) |
| 6 | // Additional URLs : https://dl.espressif.com/dl/package\_esp32\_index.json |
| 7 | // LED\_BUILTIN : Pin 2 |
| 8 | //----------------------------------------------------------------------------- |
| 9 | #include <Arduino.h> |
| 10 |  |
| 11 | #include <ESP32Servo.h> |
| 12 |  |
| 13 | #include <SPI.h> |
| 14 | #include <Wire.h> |
| 15 | #include <Adafruit\_GFX.h> |
| 16 | #include <Adafruit\_SSD1306.h> |
| 17 |  |
| 18 | #include <LiquidCrystal\_PCF8574.h> |
| 19 |  |
| 20 | #include <BLEDevice.h> |
| 21 | #include <BLEServer.h> |
| 22 | #include <BLEUtils.h> |
| 23 | #include <BLE2902.h> |
| 24 |  |
| 25 | #define PIN\_RX\_0 3 |
| 26 | #define PIN\_TX\_0 1 |
| 27 |  |
| 28 | #define I2C\_SCL 22 |
| 29 | #define I2C\_SDA 21 |
| 30 |  |
| 31 | #define PIN\_PROCESS 2 |
| 32 | #define PIN\_BTN\_START 32 |
| 33 | #define PIN\_ALARM 12 |
| 34 |  |
| 35 | #define PNP\_OFF 0 |
| 36 | #define PNP\_ON 1 |
| 37 | #define NPN\_OFF 1 |
| 38 | #define NPN\_ON 0 |
| 39 |  |
| 40 | #define EEPROM\_I2C\_ADDRESS 0x50 |
| 41 | #define PIN\_RELAY\_A 26 |
| 42 | #define PIN\_RELAY\_B 25 |
| 43 | #define PIN\_RELAY\_C 27 |
| 44 | #define PIN\_SERVO 14 |
| 45 |  |
| 46 | char buff\_str[200] = {0}; |
| 47 | unsigned long t\_old = 0; |
| 48 | int tmr\_cnt = 0; |
| 49 | int wait\_100ms = 0; |
| 50 | int alarm\_1sec = 0; |
| 51 | int error = 0; |
| 52 |  |
| 53 | int state\_ix = 0; |
| 54 | int state\_ix\_mon = 0; |
| 55 |  |
| 56 | int machine\_run = 0; |
| 57 | volatile int machine\_stop\_start\_req = 0; |
| 58 | int machine\_req\_wait\_100ms = 0; |
| 59 |  |
| 60 | int address = 0; |
| 61 | Servo myservo; |
| 62 | //----------------------------------------------------------------------------- |
| 63 | #define SCREEN\_WIDTH 128 |
| 64 | #define SCREEN\_HEIGHT 64 |
| 65 | #define OLED\_RESET 4 |
| 66 | #define GLCD\_LINE\_1 20 |
| 67 |  |
| 68 | Adafruit\_SSD1306 display(SCREEN\_WIDTH, SCREEN\_HEIGHT, &Wire, OLED\_RESET); |
| 69 |  |
| 70 | #define GLCD\_LINE\_0 5 |
| 71 | #define GLCD\_LINE\_1 20 |
| 72 | #define GLCD\_LINE\_2 35 |
| 73 | #define GLCD\_LINE\_3 50 |
| 74 |  |
| 75 | //----------------------------------------------------------------------------- |
| 76 | #define CLCD\_ADD\_27 0x27 |
| 77 | #define CLCD\_ADD\_3F 0x3F |
| 78 |  |
| 79 | #define CLCD\_LINE\_0 0 |
| 80 | #define CLCD\_LINE\_1 1 |
| 81 |  |
| 82 | LiquidCrystal\_PCF8574 clcd(CLCD\_ADD\_27); |
| 83 |  |
| 84 | //----------------------------------------------------------------------------- |
| 85 | BLEServer \*pServer; |
| 86 | BLECharacteristic \*pTxCharacteristic; |
| 87 | volatile bool bt\_connected = false; |
| 88 | volatile bool bt\_connected\_old = false; |
| 89 | String bt\_pincode\_str = ""; |
| 90 | std::string bt\_rxvalue\_str = ""; |
| 91 | String bt\_rxdata\_str = ""; |
| 92 | int bt\_login = 0; |
| 93 | String msg\_connected\_str = ""; |
| 94 | String msg\_pincode\_str = ""; |
| 95 |  |
| 96 | // See the following for generating UUIDs: |
| 97 | // https://www.uuidgenerator.net/ |
| 98 |  |
| 99 | #define SERVICE\_UUID "6E400001-B5A3-F393-E0A9-E50E24DCCA9E" |
| 100 | #define CHARACTERISTIC\_UUID\_RX "6E400002-B5A3-F393-E0A9-E50E24DCCA9E" |
| 101 | #define CHARACTERISTIC\_UUID\_TX "6E400003-B5A3-F393-E0A9-E50E24DCCA9E" |
| 102 |  |
| 103 | //----------------------------------------------------------------------------- |
| 104 | void Interrupt\_Service\_Btn\_Start(); |
| 105 |  |
| 106 | void Interrupt\_Service\_Btn\_Start() { machine\_stop\_start\_req = 1; } |
| 107 |  |
| 108 | class MyServerCallbacks: public BLEServerCallbacks { |
| 109 | void onConnect(BLEServer\* pServer) { |
| 110 | bt\_connected = true; |
| 111 | }; |
| 112 |  |
| 113 | void onDisconnect(BLEServer\* pServer) { |
| 114 | bt\_connected = false; |
| 115 | } |
| 116 | }; |
| 117 |  |
| 118 | class MyCallbacks: public BLECharacteristicCallbacks { |
| 119 | void onWrite(BLECharacteristic \*pCharacteristic) { |
| 120 | bt\_rxvalue\_str = pCharacteristic->getValue(); |
| 121 | bt\_rxdata\_str = String(bt\_rxvalue\_str.c\_str()); |
| 122 | } |
| 123 | }; |
| 124 |  |
| 125 | //----------------------------------------------------------------------------- |
| 126 | void setup() { |
| 127 | pinMode(PIN\_PROCESS, OUTPUT); |
| 128 | pinMode(PIN\_BTN\_START, INPUT\_PULLUP); |
| 129 | pinMode(PIN\_ALARM, OUTPUT); |
| 130 | pinMode(PIN\_RELAY\_A, OUTPUT); |
| 131 | pinMode(PIN\_RELAY\_B, OUTPUT); |
| 132 | pinMode(PIN\_RELAY\_C, OUTPUT); |
| 133 |  |
| 134 | digitalWrite(PIN\_PROCESS, HIGH); |
| 135 | digitalWrite(PIN\_BTN\_START, HIGH); |
| 136 | digitalWrite(PIN\_ALARM, LOW); |
| 137 | digitalWrite(PIN\_RELAY\_A, PNP\_OFF); |
| 138 | digitalWrite(PIN\_RELAY\_B, PNP\_OFF); |
| 139 | digitalWrite(PIN\_RELAY\_C, PNP\_OFF); |
| 140 |  |
| 141 | attachInterrupt(digitalPinToInterrupt(PIN\_BTN\_START) |
| 142 | , Interrupt\_Service\_Btn\_Start, FALLING); |
| 143 |  |
| 144 | sprintf(buff\_str, "Digital Lock V1"); |
| 145 |  |
| 146 | Serial.begin(115200); |
| 147 | Serial.println(buff\_str); |
| 148 |  |
| 149 | display.begin(SSD1306\_SWITCHCAPVCC, 0x3C); |
| 150 | display.setTextSize(1); |
| 151 | display.setTextColor(WHITE); |
| 152 |  |
| 153 | display.clearDisplay(); |
| 154 | display.setCursor(0, GLCD\_LINE\_0); |
| 155 | display.println(F(buff\_str)); |
| 156 | display.display(); |
| 157 |  |
| 158 | clcd.begin(16, 2); |
| 159 | clcd.init(); |
| 160 | clcd.setBacklight(255); |
| 161 | clcd.setCursor(0, CLCD\_LINE\_0); |
| 162 | clcd.print(buff\_str); |
| 163 |  |
| 164 | //----------------------------------------------------------------------------- |
| 165 | BLEDevice::init(buff\_str); |
| 166 |  |
| 167 | pServer = BLEDevice::createServer(); |
| 168 | pServer->setCallbacks(new MyServerCallbacks()); |
| 169 |  |
| 170 | BLEService \*pService = pServer->createService(SERVICE\_UUID); |
| 171 |  |
| 172 | pTxCharacteristic = pService->createCharacteristic( |
| 173 | CHARACTERISTIC\_UUID\_TX, |
| 174 | BLECharacteristic::PROPERTY\_NOTIFY |
| 175 | ); |
| 176 |  |
| 177 | pTxCharacteristic->addDescriptor(new BLE2902()); |
| 178 |  |
| 179 | BLECharacteristic \*pRxCharacteristic = pService->createCharacteristic( |
| 180 | CHARACTERISTIC\_UUID\_RX, |
| 181 | BLECharacteristic::PROPERTY\_WRITE |
| 182 | ); |
| 183 |  |
| 184 | pRxCharacteristic->setCallbacks(new MyCallbacks()); |
| 185 |  |
| 186 | pService->start(); |
| 187 |  |
| 188 | pServer->getAdvertising()->start(); |
| 189 | Serial.println("Waiting a client connection to notify..."); |
| 190 |  |
| 191 | //----------------------------------------------------------------------------- |
| 192 | t\_old = 0; |
| 193 | tmr\_cnt = 0; |
| 194 | wait\_100ms = 0; |
| 195 | alarm\_1sec = 0; |
| 196 | error = 0; |
| 197 | state\_ix = 0; |
| 198 | state\_ix\_mon = 0; |
| 199 | machine\_run = 0; |
| 200 | machine\_stop\_start\_req = 0; |
| 201 |  |
| 202 | bt\_connected = false; |
| 203 | bt\_connected\_old = false; |
| 204 | //bt\_pincode\_str = "1234"; |
| 205 | bt\_rxvalue\_str = ""; |
| 206 | bt\_login = 0; |
| 207 | msg\_connected\_str = ""; |
| 208 | msg\_pincode\_str = ""; |
| 209 |  |
| 210 | //-----------------------------------------------------------------------------EEPROM |
| 211 | while(readEEPROM(EEPROM\_I2C\_ADDRESS, address) == 30) address += 5; |
| 212 | for (int str=0; str < 4; str++) |
| 213 | bt\_pincode\_str += readEEPROM(EEPROM\_I2C\_ADDRESS, (address+str+1)); |
| 214 |  |
| 215 | //-----------------------------------------------------------------------------servo |
| 216 | ESP32PWM::allocateTimer(0); |
| 217 | ESP32PWM::allocateTimer(1); |
| 218 | ESP32PWM::allocateTimer(2); |
| 219 | ESP32PWM::allocateTimer(3); |
| 220 | myservo.setPeriodHertz(50); |
| 221 | myservo.attach(PIN\_SERVO, 1000, 2000); |
| 222 | myservo.write(10); //open130 close10 |
| 223 |  |
| 224 | Serial.print("-> code : "); |
| 225 | Serial.println(bt\_pincode\_str); |
| 226 | } |
| 227 |  |
| 228 | void loop() { |
| 229 | //----------------------------------------------------------------------------- |
| 230 | if (bt\_connected\_old != bt\_connected) { |
| 231 | bt\_connected\_old = bt\_connected; |
| 232 | if (bt\_connected == false) { |
| 233 | msg\_connected\_str = "-> Bluetooth disconnected"; |
| 234 | state\_ix = 0; |
| 235 | } else { |
| 236 | msg\_connected\_str = "-> Bluetooth connected"; |
| 237 | } |
| 238 | } |
| 239 |  |
| 240 | //----------------------------------------------------------------------------- |
| 241 | if (machine\_stop\_start\_req == 1 && bt\_connected != 1) { |
| 242 | if (machine\_req\_wait\_100ms == 0) { |
| 243 | machine\_req\_wait\_100ms = 50; |
| 244 | if(machine\_run == 1){ machine\_run = !machine\_run; } |
| 245 | } |
| 246 | } machine\_stop\_start\_req = 0; |
| 247 |  |
| 248 | //----------------------------------------------------------------------------- |
| 249 | state\_ix\_mon = state\_ix; |
| 250 |  |
| 251 | switch (state\_ix) { |
| 252 | default: |
| 253 | case 0: |
| 254 | bt\_login = 0; |
| 255 | state\_ix++; |
| 256 | break; |
| 257 |  |
| 258 | case 1: |
| 259 | wait\_100ms = 5; //<- Give the bluetooth stack the chance to get things ready |
| 260 | state\_ix++; |
| 261 | break; |
| 262 |  |
| 263 | case 2: |
| 264 | if (wait\_100ms == 0) { state\_ix++; } |
| 265 | break; |
| 266 |  |
| 267 | case 3: |
| 268 | pServer->startAdvertising(); //<- Restart advertising |
| 269 | state\_ix++; |
| 270 | break; |
| 271 |  |
| 272 | case 4: |
| 273 | if (bt\_connected == true) { state\_ix++; } |
| 274 | break; |
| 275 |  |
| 276 | case 5: |
| 277 | if (bt\_rxdata\_str.length() > 0) { |
| 278 | int bmsg\_ok = 0; |
| 279 |  |
| 280 | if (bt\_rxdata\_str.length() >= 7) { |
| 281 | String bcmd\_str = bt\_rxdata\_str.substring(0, 3); |
| 282 | String bpin\_str = bt\_rxdata\_str.substring(3, 7); |
| 283 | int bcmd\_len = bt\_rxdata\_str.length(); |
| 284 |  |
| 285 | if (bt\_rxdata\_str == "Hi, ESP32") { |
| 286 | bmsg\_ok = 1; |
| 287 | msg\_pincode\_str = "-> Hi, Flutter"; |
| 288 | } |
| 289 |  |
| 290 | if ((bcmd\_str == "C0-") && (bcmd\_len == 7) && (bpin\_str == bt\_pincode\_str)) { |
| 291 | bmsg\_ok = 1; |
| 292 | msg\_pincode\_str = "-> Command : Machine Stop"; |
| 293 | bt\_login = 0; |
| 294 | machine\_run = 0; |
| 295 | } |
| 296 |  |
| 297 | if ((bcmd\_str == "C1-") && (bcmd\_len == 7) && (bpin\_str == bt\_pincode\_str)) { |
| 298 | bmsg\_ok = 1; |
| 299 | msg\_pincode\_str = "-> Command : Machine Start"; |
| 300 | bt\_login = 1; |
| 301 | machine\_run = 1; |
| 302 | } |
| 303 |  |
| 304 | if ((bcmd\_str == "C2-") && (bcmd\_len == 7) && (bpin\_str == bt\_pincode\_str)) { |
| 305 | bmsg\_ok = 1; |
| 306 | msg\_pincode\_str = "-> Command : Login"; |
| 307 | bt\_login = 1; |
| 308 | } |
| 309 |  |
| 310 | if ((bcmd\_str == "C3-") && (bcmd\_len == 12) && (bpin\_str == bt\_pincode\_str)) { |
| 311 | bmsg\_ok = 1; |
| 312 | String bpin\_new\_str = bt\_rxdata\_str.substring(8, 12); |
| 313 | bt\_pincode\_str = bpin\_new\_str; |
| 314 | msg\_pincode\_str = "-> Command : Change Pincode to " + bpin\_new\_str; |
| 315 | bt\_login = 0; |
| 316 | //----------------------------------------------------------------------------- |
| 317 | if(readEEPROM(EEPROM\_I2C\_ADDRESS, address) == 30) |
|  | address += 5; |
| 319 | if(readEEPROM(EEPROM\_I2C\_ADDRESS, address) < 30) |
| 320 | writeEEPROM(EEPROM\_I2C\_ADDRESS, address, (readEEPROM(EEPROM\_I2C\_ADDRESS, address)+1)); |
| 321 | for (int str=0; str < 4; str++) |
| 322 | writeEEPROM(EEPROM\_I2C\_ADDRESS, (address+str+1), |
| 323 | (bt\_pincode\_str.substring(str, (str+1)).toInt())); |
| 324 | //----------------------------------------------------------------------------- |
| 325 | } |
| 326 | if ((bcmd\_str == "C4-") && (bcmd\_len == 12) && (bpin\_str == bt\_pincode\_str)) { |
| 327 | bmsg\_ok = 1; |
| 328 | String balarm\_str = bt\_rxdata\_str.substring(8, 12); |
| 329 | alarm\_1sec = balarm\_str.toInt(); |
| 330 | if (alarm\_1sec < 0) { alarm\_1sec = 0; } |
| 331 | if (alarm\_1sec > 9999) { alarm\_1sec = 9999; } |
| 332 | balarm\_str = String(alarm\_1sec); |
| 333 | msg\_pincode\_str = "-> Command : Machine Alarm " + balarm\_str + " Sec."; |
| 334 | bt\_login = 1; |
| 335 | } |
| 336 |  |
| 337 | if ((bcmd\_str == "C5-") && (bcmd\_len == 7) && (bpin\_str == bt\_pincode\_str)) { |
| 338 | bmsg\_ok = 1; |
| 339 | msg\_pincode\_str = "-> Command : Test interrupt stop start switch"; |
| 340 | bt\_login = 1; |
| 341 | machine\_stop\_start\_req = 1; |
| 342 | } |
| 343 | } |
| 344 |  |
| 345 | if (bmsg\_ok == 0) { |
| 346 | msg\_pincode\_str = "-> Invalid message!!!"; |
| 347 | } |
| 348 |  |
| 349 | bt\_rxdata\_str = ""; |
| 350 | } |
| 351 | break; |
| 352 |  |
| 353 | } |
| 354 |  |
| 355 | //----------------------------------------------------------------------------- |
| 356 | if (machine\_run == 0) { |
| 357 | digitalWrite(PIN\_RELAY\_A, PNP\_OFF); |
| 358 | digitalWrite(PIN\_RELAY\_B, PNP\_OFF); |
| 359 | myservo.write(10); //open130 close10 |
| 360 | } else { |
| 361 | digitalWrite(PIN\_RELAY\_A, PNP\_ON); |
| 362 | digitalWrite(PIN\_RELAY\_B, PNP\_ON); |
| 363 | myservo.write(130); //open130 close10 |
| 364 | } |
| 365 |  |
| 366 | if (alarm\_1sec == 0) { |
| 367 | digitalWrite(PIN\_ALARM, HIGH); |
| 368 | } else { |
| 369 | digitalWrite(PIN\_ALARM, LOW); |
| 370 | } |
| 371 |  |
| 372 | //----------------------------------------------------------------------------- |
| 373 | if (tmr\_cnt == 0) { |
| 374 | bool bbusy = false; |
| 375 |  |
| 376 | sprintf(buff\_str, " St-%02d Sw-%02d Cnt-%d Li-%d Mc-%d A-%04d | " |
| 377 | , state\_ix\_mon, machine\_req\_wait\_100ms, bt\_connected, bt\_login |
| 378 | , machine\_run, alarm\_1sec); |
| 379 | if ((bt\_connected == true) && (bt\_login == 1)) { |
| 380 | char btxdata\_str[200] = {0}; |
| 381 | sprintf(btxdata\_str, "Mc-%d A-%04d", machine\_run, alarm\_1sec); |
| 382 | pTxCharacteristic->setValue(btxdata\_str); |
| 383 | pTxCharacteristic->notify(); |
| 384 | } |
| 385 |  |
| 386 | Serial.print(buff\_str); |
| 387 |  |
| 388 | if ((bbusy == false) && (bt\_rxvalue\_str.length() > 0)) { |
| 389 | bbusy = true; |
| 390 | Serial.print("<- "); |
| 391 | for (int i = 0; i < bt\_rxvalue\_str.length(); i++) { |
| 392 | Serial.print(bt\_rxvalue\_str[i]); |
| 393 | } |
| 394 | bt\_rxvalue\_str = ""; |
| 395 | } |
| 396 |  |
| 397 | if ((bbusy == false) && (msg\_connected\_str.length() > 0)) { |
| 398 | bbusy = true; |
| 399 | Serial.print(msg\_connected\_str); deviceOLED(msg\_connected\_str); |
| 400 | msg\_connected\_str = ""; |
| 401 | } |
| 402 |  |
| 403 | if ((bbusy == false) && (msg\_pincode\_str.length() > 0)) { |
| 404 | bbusy = true; |
| 405 | Serial.print(msg\_pincode\_str); deviceOLED(msg\_pincode\_str); |
| 406 | msg\_pincode\_str = ""; |
| 407 | } |
| 408 |  |
| 409 | Serial.println(); |
| 410 | } |
| 411 |  |
| 412 | //----------------------------------------------------------------------------- |
| 413 | while ((micros() - t\_old) < 100000L); t\_old = micros(); |
| 414 | tmr\_cnt++; if (tmr\_cnt >= 10) { tmr\_cnt = 0; } |
| 415 |  |
| 416 | if (wait\_100ms > 0) { wait\_100ms--; } |
| 417 |  |
| 418 | if (tmr\_cnt == 0) { |
| 419 | if (alarm\_1sec > 0) { alarm\_1sec--; } |
| 420 | } |
| 421 |  |
| 422 | if (machine\_req\_wait\_100ms > 0) { machine\_req\_wait\_100ms--; } |
| 423 | } |
| 424 |  |
| 425 | //---------------------------------------------------------------------------- |
| 426 | void writeEEPROM(int deviceaddress, |
| 427 | unsigned int eeaddress, byte data ) |
| 428 | { |
| 429 | Wire.beginTransmission(deviceaddress); |
| 430 | Wire.write((int)(eeaddress >> 8)); // MSB |
| 431 | Wire.write((int)(eeaddress & 0xFF)); // LSB |
| 432 | Wire.write(data); |
| 433 | Wire.endTransmission(); |
| 434 | delay(5); |
| 435 | } |
| 436 |  |
| 437 | byte readEEPROM(int deviceaddress, unsigned int eeaddress ) |
| 438 | { |
| 439 | byte rdata = 0xFF; |
| 440 |  |
| 441 | Wire.beginTransmission(deviceaddress); |
| 442 | Wire.write((int)(eeaddress >> 8)); // MSB |
| 443 | Wire.write((int)(eeaddress & 0xFF)); // LSB |
| 444 | Wire.endTransmission(); |
| 445 | Wire.requestFrom(deviceaddress,1); |
| 446 |  |
| 447 | if (Wire.available()) rdata = Wire.read(); |
| 448 | return rdata; |
| 449 | } |
| 450 | void deviceOLED(String str){ |
| 451 | char p[str.length()]; |
| 452 | int i; for (i = 0; i < sizeof(p); i++) |
| 453 | p[i] = str[i]; |
| 454 | p[i] = '\0'; |
| 455 | display.clearDisplay(); |
| 456 | display.setCursor(0, GLCD\_LINE\_1); |
| 457 | display.println(F(p)); |
| 458 | display.display(); |
| 459 | } |