**ภาคผนวก จ**

โปรแกรมการทำงาน

โปรแกรมอาร์ดูย์โน่ไอดีอีบอร์ด Node 32 Lite

|  |  |
| --- | --- |
| 1 | #include <WiFi.h> |
| 2 | #include <FirebaseESP32.h> |
| 3 |  |
| 4 | #define WIFI\_SSID "piyakul" |
| 5 | #define WIFI\_PASSWORD "06052539" |
| 6 | #define FIREBASE\_HOST "moarduino-c3464.firebaseio.com" |
| 7 | #define FIREBASE\_AUTH "CSPix239KS3OZfSLzMmQs19FelUCFcQpG5lyGecc" |
| 8 |  |
| 9 | FirebaseData firebaseData\_Rx; |
| 10 | FirebaseData firebaseData\_Tx; |
| 11 |  |
| 12 | String system\_path = "/Line\_01/System"; |
| 13 | String log\_path = "/Line\_01/Log"; |
| 14 |  |
| 15 | char TESTMO[32]; |
| 16 |  |
| 17 | #define NPN\_ON 0 |
| 18 | #define NPN\_OFF 1 |
| 19 | #define DIR\_FW 1 |
| 20 | #define DIR\_RW 0 |
| 21 |  |
| 22 | #define PROC\_INITIAL 0 |
| 23 | #define PROC\_STANDBY 1 |
| 24 | #define PROC\_COMPLETED 3 |
| 25 | #define PROC\_ERROR 6 |
| 26 | #define PROC\_PRODUCTION\_1 11 |
| 27 | #define PROC\_PRODUCTION\_2 12 |
| 28 | #define PROC\_PRODUCTION\_3 13 |
| 29 | #define PROC\_PRODUCTION\_4 14 |
| 30 | #define PROC\_PRODUCTION\_5 15 |
| 31 | #define PROC\_PRODUCTION\_6 16 |
| 32 | #define PROC\_PRODUCTION\_7 17 |
| 33 |  |
| 34 | #define PIN\_MOTOR\_ON 16 |
| 35 | #define PIN\_MOTOR\_DIR 17 |
| 36 | #define PIN\_SENSOR\_0 4 |
| 37 | #define PIN\_SENSOR\_1 0 |
| 38 | #define PIN\_SENSOR\_2 2 |
| 39 | #define PIN\_SENSOR\_3 13 |
| 40 |  |
| 41 | #define PIN\_RX 3 |
| 42 | #define PIN\_TX 1 |
| 43 |  |
| 44 | #define TMR\_DWN\_MAX 200 |
| 45 |  |
| 46 | char str\_buff[200] = {0}; |
| 47 | unsigned long t\_old = 0; |
| 48 | int tmr\_dwn = 0; |
| 49 | int tmr\_cnt = 0; |
| 50 | int pin\_sensor[4] = {0}; |
| 51 | int sensor[4] = {0}; |
| 52 | int error = 0; |
| 53 |  |
| 54 | int status\_robot\_1 = 0; |
| 55 | int status\_robot\_2 = 0; |
| 56 | int status\_robot\_3 = 0; |
| 57 | int status\_mobile = 0; |
| 58 | int status\_mobile\_mon = 0; |
| 59 | int status\_process = 0; |
| 60 | int status\_process\_mon = 0; |
| 61 | int status\_product = 0; |
| 62 | int status\_product\_mon = 0; |
| 63 | int status\_conveyer = 0; |
| 64 | int status\_conveyer\_mon = 0; |
| 65 |  |
| 66 | String mobile\_qrcode = ""; |
| 67 | String product\_qrcode = ""; |
| 68 | //------------------------------------------------------ |
| 69 | #include <WiFiUdp.h> |
| 70 |  |
| 71 | unsigned int udp\_localPort = 1000; |
| 72 |  |
| 73 | IPAddress timeServerIP; // time.nist.gov NTP server address |
| 74 | const char\* ntpServerName = "time.nist.gov"; |
| 75 | const int NTP\_PACKET\_SIZE = 48; // NTP time stamp is in the first 48 bytes of the message |
| 76 | byte packetBuffer[ NTP\_PACKET\_SIZE]; // buffer to hold incoming and outgoing packets |
| 77 |  |
| 78 | WiFiUDP udp; |
| 79 |  |
| 80 | int ntp\_Y, ntp\_M, ntp\_D, ntp\_H, ntp\_m, ntp\_S; |
| 81 |  |
| 82 | unsigned long sendNTPpacket(IPAddress& address); |
| 83 | //------------------------------------------------------ |
| 84 | void Motor\_Stop(); |
| 85 | void Motor\_FW(); |
| 86 | void Motor\_RW(); |
| 87 |  |
| 88 | void streamCallback(StreamData data); |
| 89 | void streamTimeoutCallback(bool ptimeout); |
| 90 | int System\_Update(String pitem, String pdespt); |
| 91 | int System\_Update\_Mobile(int pstatus); |
| 92 | int System\_Update\_Process(int pstatus); |
| 93 | int System\_Update\_Product(int pstatus); |
| 94 | int System\_Update\_Conveyer(int pstatus); |
| 95 | int Log\_Add(String pdatetime, String pdetail); |
| 96 |  |
| 97 | void printResult(FirebaseData &data); |
| 98 | void printResult(StreamData &data); |
| 99 |  |
| 100 | void setup() |
| 101 | { |
| 102 | pinMode(PIN\_MOTOR\_ON, OUTPUT); |
| 103 | pinMode(PIN\_MOTOR\_DIR, OUTPUT); |
| 104 |  |
| 105 | pin\_sensor[0] = PIN\_SENSOR\_0; |
| 106 | pin\_sensor[1] = PIN\_SENSOR\_1; |
| 107 | pin\_sensor[2] = PIN\_SENSOR\_2; |
| 108 | pin\_sensor[3] = PIN\_SENSOR\_3; |
| 109 |  |
| 110 | for (int bix = 0; bix < 4; bix++) { |
| 111 | pinMode(pin\_sensor[bix], INPUT\_PULLUP); |
| 112 | } |
| 113 |  |
| 114 | Motor\_Stop(); |
| 115 |  |
| 116 | Serial1.begin(9600, SERIAL\_8N1, PIN\_RX, PIN\_TX); |
| 117 |  |
| 118 | WiFi.begin(WIFI\_SSID, WIFI\_PASSWORD); |
| 119 | Serial1.print("\nConnecting to Wi-Fi"); |
| 120 | while (WiFi.status() != WL\_CONNECTED) { |
| 121 | Serial1.print("."); |
| 122 | delay(300); |
| 123 | } |
| 124 |  |
| 125 | Serial1.print("\nConnected with IP: "); |
| 126 | Serial1.println(WiFi.localIP()); |
| 127 |  |
| 128 | Firebase.begin(FIREBASE\_HOST, FIREBASE\_AUTH); |
| 129 | Firebase.reconnectWiFi(true); |
| 130 |  |
| 131 | if (!Firebase.beginStream(firebaseData\_Rx, system\_path)) { |
| 132 | Serial1.println("------------------------------------"); |
| 133 | Serial1.println("Can't begin stream connection..."); |
| 134 | Serial1.println("REASON: " + firebaseData\_Rx.errorReason()); |
| 135 | Serial1.println("------------------------------------"); |
| 136 | Serial1.println(); |
| 137 | } |
| 138 |  |
| 139 | Firebase.setStreamCallback(firebaseData\_Rx, streamCallback, streamTimeoutCallback); |
| 140 |  |
| 141 | error = 0; |
| 142 | error += System\_Update("Robot\_1", "Initial"); |
| 143 | error += System\_Update("Robot\_2", "Initial"); |
| 144 | error += System\_Update("Robot\_3", "Initial"); |
| 145 | error += System\_Update("Mobile", "Initial"); |
| 146 | error += System\_Update("Process", "Initial"); |
| 147 | error += System\_Update("Product", "No Product"); |
| 148 | error += System\_Update("Conveyer", "Stop"); |
| 149 | if (error > 0) { status\_process = PROC\_ERROR; } |
| 150 | delay(200); |
| 151 | //------------------------------------------------------ |
| 152 | udp.begin(udp\_localPort); |
| 153 | ntp\_Y = 0; ntp\_M = 0; ntp\_D = 0; |
| 154 | ntp\_H = 0; ntp\_m = 0; ntp\_S = 0; |
| 155 | } |
| 156 |  |
| 157 | void loop() |
| 158 | { |
| 159 | for (int bix = 0; bix < 4; bix++) { |
| 160 | sensor[bix] = !digitalRead(pin\_sensor[bix]); |
| 161 | } |
| 162 |  |
| 163 | status\_product = 0; |
| 164 | if (tmr\_dwn > 0) { status\_product = 1; } |
| 165 |  |
| 166 | if (status\_process\_mon != status\_process) { |
| 167 | status\_process\_mon = status\_process; |
| 168 | System\_Update\_Process(status\_process\_mon); |
| 169 | } |
| 170 |  |
| 171 | if (status\_product\_mon != status\_product) { |
| 172 | status\_product\_mon = status\_product; |
| 173 | System\_Update\_Product(status\_product\_mon); |
| 174 | } |
| 175 |  |
| 176 | if (status\_conveyer\_mon != status\_conveyer) { |
| 177 | status\_conveyer\_mon = status\_conveyer; |
| 178 | System\_Update\_Conveyer(status\_conveyer\_mon); |
| 179 | } |
| 180 |  |
| 181 | switch (status\_process) { |
| 182 | default: |
| 183 | status\_process = PROC\_INITIAL; |
| 184 | case PROC\_INITIAL: |
| 185 | Motor\_Stop(); status\_conveyer = 0; |
| 186 |  |
| 187 | status\_product = 0; |
| 188 | mobile\_qrcode = ""; |
| 189 | product\_qrcode = ""; |
| 190 |  |
| 191 | error = 0; |
| 192 | error += System\_Update("Robot\_1", "Close"); |
| 193 | error += System\_Update("Mobile", "Initial"); |
| 194 |  |
| 195 | if (error > 0) { |
| 196 | status\_process = PROC\_ERROR; |
| 197 | } else { |
| 198 | status\_process = PROC\_STANDBY; |
| 199 | } |
| 200 | break; |
| 201 |  |
| 202 | case PROC\_STANDBY: |
| 203 | Motor\_Stop(); status\_conveyer = 0; |
| 204 |  |
| 205 | if (sensor[0] == 1) { |
| 206 | tmr\_dwn = TMR\_DWN\_MAX; |
| 207 | } |
| 208 |  |
| 209 | if (sensor[0] == 1 |
| 210 | && sensor[1] == 0 |
| 211 | && sensor[2] == 0 |
| 212 | && sensor[3] == 0) { |
| 213 | error = System\_Update("Robot\_1", "Open"); |
| 214 | status\_process = PROC\_PRODUCTION\_1; |
| 215 | } |
| 216 | break; |
| 217 |  |
| 218 | case PROC\_PRODUCTION\_1: |
| 219 | Motor\_FW(); status\_conveyer = 1; |
| 220 |  |
| 221 | if (sensor[0] == 0 |
| 222 | && sensor[1] == 1 |
| 223 | && sensor[2] == 0 |
| 224 | && sensor[3] == 0) { |
| 225 | Motor\_Stop(); status\_conveyer = 0; |
| 226 | error = System\_Update("Robot\_1", "Close"); |
| 227 | error = System\_Update("Mobile", "QR Request"); |
| 228 | status\_process = PROC\_PRODUCTION\_2; |
| 229 | } |
| 230 | break; |
| 231 |  |
| 232 | case PROC\_PRODUCTION\_2: |
| 233 | Motor\_Stop(); status\_conveyer = 0; |
| 234 |  |
| 235 | if (status\_mobile == 4) { |
| 236 | Motor\_FW(); status\_conveyer = 1; |
| 237 | product\_qrcode = mobile\_qrcode; |
| 238 | status\_process = PROC\_PRODUCTION\_3; |
| 239 | } |
| 240 | break; |
| 241 |  |
| 242 | case PROC\_PRODUCTION\_3: |
| 243 | Motor\_FW(); status\_conveyer = 1; |
| 244 |  |
| 245 | if (sensor[2] == 1 |
| 246 | || sensor[3] == 1) { |
| 247 | Motor\_Stop(); status\_conveyer = 0; |
| 248 | status\_process = PROC\_PRODUCTION\_4; |
| 249 | } |
| 250 |  |
| 251 | if (sensor[0] == 1) { |
| 252 | Motor\_Stop(); status\_conveyer = 0; |
| 253 | } else { |
| 254 | if (tmr\_dwn > 0) { |
| 255 | tmr\_dwn--; |
| 256 | } else { |
| 257 | Motor\_Stop(); status\_conveyer = 0; |
| 258 | status\_process = PROC\_INITIAL; |
| 259 | } |
| 260 | } |
| 261 | break; |
| 262 |  |
| 263 | case PROC\_PRODUCTION\_4: |
| 264 | Motor\_Stop(); status\_conveyer = 0; |
| 265 | // Log & Send data to Poom & Sam robot |
| 266 | if(product\_qrcode == "P01"){ |
| 267 | Serial1.print("1"); |
| 268 | Serial1.print("\n"); |
| 269 | delay(1000); |
| 270 | } |
| 271 |  |
| 272 | if(product\_qrcode == "P02"){ |
| 273 | Serial1.print("2"); |
| 274 | Serial1.print("\n"); |
| 275 | delay(1000); |
| 276 | } |
| 277 |  |
| 278 | if(product\_qrcode == "P03"){ |
| 279 | Serial1.print("3"); |
| 280 | Serial1.print("\n"); |
| 281 | delay(1000); |
| 282 | } |
| 283 |  |
| 284 | if(product\_qrcode == "P04"){ |
| 285 | Serial1.print("4"); |
| 286 | Serial1.print("\n"); |
| 287 | delay(1000); |
| 288 | } |
| 289 |  |
| 290 | sprintf(str\_buff, "%04d-%02d-%02d %02d:%02d:%02d", ntp\_Y, ntp\_M, ntp\_D, ntp\_H, ntp\_m, ntp\_S); |
| 291 | error = Log\_Add(str\_buff, product\_qrcode); |
| 292 |  |
| 293 | status\_process = PROC\_PRODUCTION\_5; |
| 294 | break; |
| 295 |  |
| 296 | case PROC\_PRODUCTION\_5: |
| 297 | Motor\_Stop(); status\_conveyer = 0; |
| 298 |  |
| 299 | if (sensor[0] == 0 |
| 300 | && sensor[1] == 0 |
| 301 | && sensor[2] == 0 |
| 302 | && sensor[3] == 0) { |
| 303 | status\_process = PROC\_INITIAL; |
| 304 | } |
| 305 | break; |
| 306 |  |
| 307 | case PROC\_PRODUCTION\_6: break; |
| 308 | case PROC\_PRODUCTION\_7: break; |
| 309 | case PROC\_COMPLETED: break; |
| 310 |  |
| 311 | case PROC\_ERROR: |
| 312 | Motor\_Stop(); |
| 313 | break; |
| 314 | } |
| 315 |  |
| 316 | tmr\_cnt++; |
| 317 | if (tmr\_cnt == 10) { |
| 318 | tmr\_cnt = 0; |
| 319 |  |
| 320 | int bbytes = udp.parsePacket(); |
| 321 |  |
| 322 | if (bbytes > 0) { |
| 323 | } else { |
| 324 | udp.read(packetBuffer, NTP\_PACKET\_SIZE); |
| 325 |  |
| 326 | unsigned long highWord = word(packetBuffer[40], packetBuffer[41]); |
| 327 | unsigned long lowWord = word(packetBuffer[42], packetBuffer[43]); |
| 328 | unsigned long secsSince1900 = highWord << 16 | lowWord; |
| 329 | unsigned long secs2020 = 3786825600UL; |
| 330 |  |
| 331 | if (secsSince1900 > 0) { |
| 332 | unsigned long epoch = secsSince1900 - secs2020; |
| 333 | unsigned long gmt\_offset = +7UL; |
| 334 | epoch = (epoch + (gmt\_offset \* 3600UL)); |
| 335 |  |
| 336 | ntp\_Y = (int)2020; |
| 337 | ntp\_M = (int)3; |
| 338 | ntp\_D = (int)(((epoch / 86400UL) + 1UL) - 60); |
| 339 |  |
| 340 | ntp\_H = (int)((epoch % 86400UL) / 3600UL); |
| 341 | ntp\_m = (int)((epoch % 3600UL) / 60UL); |
| 342 | ntp\_S = (int) (epoch % 60UL); |
| 343 | } |
| 344 | } |
| 345 |  |
| 346 | WiFi.hostByName(ntpServerName, timeServerIP); |
| 347 | sendNTPpacket(timeServerIP); |
| 348 |  |
| 349 | sprintf(str\_buff, "[ %02d | %d | %d%d%d%d | %d | %2d.%1d ] [ %04d-%02d-%02d %02d:%02d:%02d ] [ " |
| 350 | , status\_process\_mon |
| 351 | , status\_conveyer\_mon |
| 352 | , sensor[0], sensor[1], sensor[2], sensor[3] |
| 353 | , status\_product\_mon |
| 354 | , (tmr\_dwn / 10), (tmr\_dwn % 10) |
| 355 | , ntp\_Y, ntp\_M, ntp\_D, ntp\_H, ntp\_m, ntp\_S |
| 356 | ); |
| 357 | Serial1.println(str\_buff + product\_qrcode + " ] "); |
| 358 | } |
| 359 |  |
| 360 | while ((micros() - t\_old) < 100000L); |
| 361 | t\_old = micros(); |
| 362 |  |
| 363 | } |
| 364 |  |
| 365 | void Motor\_Stop() { |
| 366 | digitalWrite(PIN\_MOTOR\_ON, NPN\_OFF); |
| 367 | digitalWrite(PIN\_MOTOR\_DIR, DIR\_FW); |
| 368 | } |
| 369 |  |
| 370 | void Motor\_FW() { |
| 371 | digitalWrite(PIN\_MOTOR\_ON, NPN\_ON); |
| 372 | digitalWrite(PIN\_MOTOR\_DIR, DIR\_FW); |
| 373 | } |
| 374 |  |
| 375 | void Motor\_RW() { |
| 376 | digitalWrite(PIN\_MOTOR\_ON, NPN\_ON); |
| 377 | digitalWrite(PIN\_MOTOR\_DIR, DIR\_RW); |
| 378 | } |
| 379 |  |
| 380 | void streamCallback(StreamData data) { |
| 381 |  |
| 382 | if (data.dataPath() == "/Mobile/status") { |
| 383 | if (data.dataType() == "int") { status\_mobile = data.intData(); } |
| 384 | } |
| 385 |  |
| 386 | if (data.dataPath() == "/Mobile/despt") { |
| 387 | if (data.dataType() == "string") { mobile\_qrcode = data.stringData(); } |
| 388 | } |
| 389 |  |
| 390 | if (data.dataPath() == "/Mobile") { |
| 391 |  |
| 392 | String bmobile\_despt = ""; |
| 393 | String bmobile\_status = ""; |
| 394 |  |
| 395 | FirebaseJson \*json = data.jsonObjectPtr(); |
| 396 |  |
| 397 | size\_t len = json->iteratorBegin(); |
| 398 | String key, value = ""; |
| 399 | int type = 0; |
| 400 | for (size\_t i = 0; i < len; i++) |
| 401 | { |
| 402 | json->iteratorGet(i, type, key, value); |
| 403 |  |
| 404 | if (key == "status") { |
| 405 | bmobile\_status = value; |
| 406 | } |
| 407 |  |
| 408 | if (key == "despt") { |
| 409 | bmobile\_despt = value; |
| 410 | } |
| 411 | } |
| 412 | json->iteratorEnd(); |
| 413 |  |
| 414 | if (bmobile\_status == "4") { |
| 415 |  |
| 416 | status\_mobile = 4; |
| 417 | mobile\_qrcode = bmobile\_despt; |
| 418 | } |
| 419 | } |
| 420 | } |
| 421 |  |
| 422 | void streamTimeoutCallback(bool ptimeout) { |
| 423 | if (ptimeout != 0) { Serial1.println("Stream timeout, resume streaming..."); } |
| 424 | } |
| 425 |  |
| 426 | int System\_Update(String pitem, String pdespt) { |
| 427 | int lresult\_int = 0; |
| 428 | int lstatus = 6; |
| 429 | String ldespt = pdespt; |
| 430 |  |
| 431 | if (pitem == "Robot\_1") { |
| 432 | if (pdespt == "Initial") { lstatus = 0; } |
| 433 | if (pdespt == "Standby") { lstatus = 1; } |
| 434 | if (pdespt == "Open") { lstatus = 2; } |
| 435 | if (pdespt == "Close") { lstatus = 3; } |
| 436 | status\_robot\_1 = lstatus; |
| 437 | } |
| 438 |  |
| 439 | if (pitem == "Robot\_2") { |
| 440 | if (pdespt == "Initial") { lstatus = 0; } |
| 441 | status\_robot\_2 = lstatus; |
| 442 | } |
| 443 |  |
| 444 | if (pitem == "Robot\_3") { |
| 445 | if (pdespt == "Initial") { lstatus = 0; } |
| 446 | status\_robot\_3 = lstatus; |
| 447 | } |
| 448 |  |
| 449 | if (pitem == "Mobile") { |
| 450 | if (pdespt == "Initial") { lstatus = 0; } |
| 451 | if (pdespt == "Mobile Check") { lstatus = 1; } |
| 452 | if (pdespt == "Mobile Ready") { lstatus = 2; } |
| 453 | if (pdespt == "QR Request") { lstatus = 3; } |
| 454 | if (pdespt == "QR Completed") { lstatus = 4; } |
| 455 | status\_mobile = lstatus; |
| 456 | } |
| 457 |  |
| 458 | if (pitem == "Process") { |
| 459 | if (pdespt == "Initial") { lstatus = 0; } |
| 460 | if (pdespt == "Standby") { lstatus = 1; } |
| 461 | if (pdespt == "Completed") { lstatus = 3; } |
| 462 | if (pdespt == "Production\_1") { lstatus = 11; } |
| 463 | if (pdespt == "Production\_2") { lstatus = 12; } |
| 464 | if (pdespt == "Production\_3") { lstatus = 13; } |
| 465 | if (pdespt == "Production\_4") { lstatus = 14; } |
| 466 | if (pdespt == "Production\_5") { lstatus = 15; } |
| 467 | if (pdespt == "Production\_6") { lstatus = 16; } |
| 468 | if (pdespt == "Production\_7") { lstatus = 17; } |
| 469 | status\_process = lstatus; |
| 470 | } |
| 471 |  |
| 472 | if (pitem == "Product") { |
| 473 | if (pdespt == "No Product") { lstatus = 0; } |
| 474 | if (pdespt == "Product In Line") { lstatus = 1; } |
| 475 | status\_product = lstatus; |
| 476 | } |
| 477 |  |
| 478 | if (pitem == "Conveyer") { |
| 479 | if (pdespt == "Stop") { lstatus = 0; } |
| 480 | if (pdespt == "Run") { lstatus = 1; } |
| 481 | status\_conveyer = lstatus; |
| 482 | } |
| 483 |  |
| 484 | if (lstatus == 6) { ldespt = "Error"; } |
| 485 |  |
| 486 | FirebaseJson json; |
| 487 | json.add("status", lstatus).add("despt", ldespt); |
| 488 | if (Firebase.setJSON(firebaseData\_Tx, system\_path + "/" + pitem, json) != 0) { |
| 489 | } else { |
| 490 | lresult\_int = 1; |
| 491 | Serial1.println("Tx FAILED: REASON: " + firebaseData\_Tx.errorReason()); |
| 492 | } |
| 493 |  |
| 494 | return lresult\_int; |
| 495 | } |
| 496 |  |
| 497 | int System\_Update\_Mobile(int pstatus) { |
| 498 | int lresult\_int = 0; |
| 499 | String ldespt = "Error"; |
| 500 |  |
| 501 | if (pstatus == 0) { ldespt = "Initial"; } |
| 502 | if (pstatus == 1) { ldespt = "Mobile Check"; } |
| 503 | if (pstatus == 2) { ldespt = "Mobile Ready"; } |
| 504 | if (pstatus == 3) { ldespt = "QR Request"; } |
| 505 | if (pstatus == 4) { ldespt = "QR Completed"; } |
| 506 |  |
| 507 | lresult\_int = System\_Update("Mobile", ldespt); |
| 508 |  |
| 509 | return lresult\_int; |
| 510 | } |
| 511 |  |
| 512 | int System\_Update\_Process(int pstatus) { |
| 513 | int lresult\_int = 0; |
| 514 | String ldespt = "Error"; |
| 515 |  |
| 516 | if (pstatus == 0) { ldespt = "Initial"; } |
| 517 | if (pstatus == 1) { ldespt = "Standby"; } |
| 518 | if (pstatus == 3) { ldespt = "Completed"; } |
| 519 | if (pstatus == 11) { ldespt = "Production\_1"; } |
| 520 | if (pstatus == 12) { ldespt = "Production\_2"; } |
| 521 | if (pstatus == 13) { ldespt = "Production\_3"; } |
| 522 | if (pstatus == 14) { ldespt = "Production\_4"; } |
| 523 | if (pstatus == 15) { ldespt = "Production\_5"; } |
| 524 | if (pstatus == 16) { ldespt = "Production\_6"; } |
| 525 | if (pstatus == 17) { ldespt = "Production\_7"; } |
| 526 |  |
| 527 | lresult\_int = System\_Update("Process", ldespt); |
| 528 |  |
| 529 | return lresult\_int; |
| 530 | } |
| 531 |  |
| 532 | int System\_Update\_Product(int pstatus) { |
| 533 | int lresult\_int = 0; |
| 534 | String ldespt = "Error"; |
| 535 |  |
| 536 | if (pstatus == 0) { ldespt = "No Product"; } |
| 537 | if (pstatus == 1) { ldespt = "Product In Line"; } |
| 538 |  |
| 539 | lresult\_int = System\_Update("Product", ldespt); |
| 540 |  |
| 541 | return lresult\_int; |
| 542 | } |
| 543 |  |
| 544 | int System\_Update\_Conveyer(int pstatus) { |
| 545 | int lresult\_int = 0; |
| 546 | String ldespt = "Error"; |
| 547 |  |
| 548 | if (pstatus == 0) { ldespt = "Stop"; } |
| 549 | if (pstatus == 1) { ldespt = "Run"; } |
| 550 |  |
| 551 | lresult\_int = System\_Update("Conveyer", ldespt); |
| 552 |  |
| 553 | return lresult\_int; |
| 554 | } |
| 555 |  |
| 556 | int Log\_Add(String pdatetime, String pdetail) { |
| 557 | int lresult\_int = 0; |
| 558 | String llog\_name = pdatetime + " " + pdetail; |
| 559 | FirebaseJson json; |
| 560 | json.add("datetime", pdatetime).add("detail", pdetail); |
| 561 | if (Firebase.setJSON(firebaseData\_Tx, log\_path + "/" + llog\_name, json) != 0) { |
| 562 | } else { |
| 563 | lresult\_int = 1; |
| 564 | Serial1.println("Tx FAILED: REASON: " + firebaseData\_Tx.errorReason()); |
| 565 | } |
| 566 | return lresult\_int; |
| 567 | } |
| 568 |  |
| 569 | void printResult(FirebaseData &data) |
| 570 | { |
| 571 | Serial1.println("FirebaseData"); |
| 572 |  |
| 573 | if (data.dataType() == "int") |
| 574 | Serial1.println(data.intData()); |
| 575 | else if (data.dataType() == "float") |
| 576 | Serial1.println(data.floatData(), 5); |
| 577 | else if (data.dataType() == "double") |
| 578 | printf("%.9lf\n", data.doubleData()); |
| 579 | else if (data.dataType() == "boolean") |
| 580 | Serial1.println(data.boolData() == 1 ? "true" : "false"); |
| 581 | else if (data.dataType() == "string") |
| 582 | Serial1.println(data.stringData()); |
| 583 | else if (data.dataType() == "json") |
| 584 | { |
| 585 | Serial1.println(); |
| 586 | FirebaseJson &json = data.jsonObject(); |
| 587 | //Print all object data |
| 588 | Serial1.println("Pretty printed JSON data:"); |
| 589 | String jsonStr; |
| 590 | json.toString(jsonStr, true); |
| 591 | Serial1.println(jsonStr); |
| 592 | Serial1.println(); |
| 593 | Serial1.println("Iterate JSON data:"); |
| 594 | Serial1.println(); |
| 595 | size\_t len = json.iteratorBegin(); |
| 596 | String key, value = ""; |
| 597 | int type = 0; |
| 598 | for (size\_t i = 0; i < len; i++) |
| 599 | { |
| 600 | json.iteratorGet(i, type, key, value); |
| 601 | Serial1.print(i); |
| 602 | Serial1.print(", "); |
| 603 | Serial1.print("Type: "); |
| 604 | Serial1.print(type == JSON\_OBJECT ? "object" : "array"); |
| 605 | if (type == JSON\_OBJECT) |
| 606 | { |
| 607 | Serial1.print(", Key: "); |
| 608 | Serial1.print(key); |
| 609 | } |
| 610 | Serial1.print(", Value: "); |
| 611 | Serial1.println(value); |
| 612 | } |
| 613 | json.iteratorEnd(); |
| 614 | } |
| 615 | else if (data.dataType() == "array") |
| 616 | { |
| 617 | Serial1.println(); |
| 618 | //get array data from FirebaseData using FirebaseJsonArray object |
| 619 | FirebaseJsonArray &arr = data.jsonArray(); |
| 620 | //Print all array values |
| 621 | Serial1.println("Pretty printed Array:"); |
| 622 | String arrStr; |
| 623 | arr.toString(arrStr, true); |
| 624 | Serial1.println(arrStr); |
| 625 | Serial1.println(); |
| 626 | Serial1.println("Iterate array values:"); |
| 627 | Serial1.println(); |
| 628 | for (size\_t i = 0; i < arr.size(); i++) |
| 629 | { |
| 630 | Serial1.print(i); |
| 631 | Serial1.print(", Value: "); |
| 632 |  |
| 633 | FirebaseJsonData &jsonData = data.jsonData(); |
| 634 | //Get the result data from FirebaseJsonArray object |
| 635 | arr.get(jsonData, i); |
| 636 | if (jsonData.typeNum == JSON\_BOOL) |
| 637 | Serial1.println(jsonData.boolValue ? "true" : "false"); |
| 638 | else if (jsonData.typeNum == JSON\_INT) |
| 639 | Serial1.println(jsonData.intValue); |
| 640 | else if (jsonData.typeNum == JSON\_DOUBLE) |
| 641 | printf("%.9lf\n", jsonData.doubleValue); |
| 642 | else if (jsonData.typeNum == JSON\_STRING || |
| 643 | jsonData.typeNum == JSON\_NULL || |
| 644 | jsonData.typeNum == JSON\_OBJECT || |
| 645 | jsonData.typeNum == JSON\_ARRAY) |
| 646 | Serial1.println(jsonData.stringValue); |
| 647 | } |
| 648 | } |
| 649 | } |
| 650 |  |
| 651 | void printResult(StreamData &data) |
| 652 | { |
| 653 | Serial1.println("StreamData"); |
| 654 |  |
| 655 | if (data.dataType() == "int") |
| 656 | Serial1.println(data.intData()); |
| 657 | else if (data.dataType() == "float") |
| 658 | Serial1.println(data.floatData(), 5); |
| 659 | else if (data.dataType() == "double") |
| 660 | printf("%.9lf\n", data.doubleData()); |
| 661 | else if (data.dataType() == "boolean") |
| 662 | Serial1.println(data.boolData() == 1 ? "true" : "false"); |
| 663 | else if (data.dataType() == "string") |
| 664 | Serial1.println(data.stringData()); |
| 665 | else if (data.dataType() == "json") |
| 666 | { |
| 667 | Serial1.println(); |
| 668 | FirebaseJson \*json = data.jsonObjectPtr(); |
| 669 | //Print all object data |
| 670 | Serial1.println("Pretty printed JSON data:"); |
| 671 | String jsonStr; |
| 672 | json->toString(jsonStr, true); |
| 673 | Serial1.println(jsonStr); |
| 674 | Serial1.println(); |
| 675 | Serial1.println("Iterate JSON data:"); |
| 676 | Serial1.println(); |
| 677 | size\_t len = json->iteratorBegin(); |
| 678 | String key, value = ""; |
| 679 | int type = 0; |
| 680 | for (size\_t i = 0; i < len; i++) |
| 681 | { |
| 682 | json->iteratorGet(i, type, key, value); |
| 683 | Serial1.print(i); |
| 684 | Serial1.print(", "); |
| 685 | Serial1.print("Type: "); |
| 686 | Serial1.print(type == JSON\_OBJECT ? "object" : "array"); |
| 687 | if (type == JSON\_OBJECT) |
| 688 | { |
| 689 | Serial1.print(", Key: "); |
| 690 | Serial1.print(key); |
| 691 | } |
| 692 | Serial1.print(", Value: "); |
| 693 | Serial1.println(value); |
| 694 | } |
| 695 | json->iteratorEnd(); |
| 696 | } |
| 697 | else if (data.dataType() == "array") |
| 698 | { |
| 699 | Serial1.println(); |
| 700 | //get array data from FirebaseData using FirebaseJsonArray object |
| 701 | FirebaseJsonArray \*arr = data.jsonArrayPtr(); |
| 702 | //Print all array values |
| 703 | Serial1.println("Pretty printed Array:"); |
| 704 | String arrStr; |
| 705 | arr->toString(arrStr, true); |
| 706 | Serial1.println(arrStr); |
| 707 | Serial1.println(); |
| 708 | Serial1.println("Iterate array values:"); |
| 709 | Serial1.println(); |
| 710 |  |
| 711 | for (size\_t i = 0; i < arr->size(); i++) |
| 712 | { |
| 713 | Serial1.print(i); |
| 714 | Serial1.print(", Value: "); |
| 715 |  |
| 716 | FirebaseJsonData \*jsonData = data.jsonDataPtr(); |
| 717 | //Get the result data from FirebaseJsonArray object |
| 718 | arr->get(\*jsonData, i); |
| 719 | if (jsonData->typeNum == JSON\_BOOL) |
| 720 | Serial1.println(jsonData->boolValue ? "true" : "false"); |
| 721 | else if (jsonData->typeNum == JSON\_INT) |
| 722 | Serial1.println(jsonData->intValue); |
| 723 | else if (jsonData->typeNum == JSON\_DOUBLE) |
| 724 | printf("%.9lf\n", jsonData->doubleValue); |
| 725 | else if (jsonData->typeNum == JSON\_STRING || |
| 726 | jsonData->typeNum == JSON\_NULL || |
| 727 | jsonData->typeNum == JSON\_OBJECT || |
| 728 | jsonData->typeNum == JSON\_ARRAY) |
| 729 | Serial1.println(jsonData->stringValue); |
| 730 | } |
| 731 | } |
| 732 | } |
| 733 |  |
| 734 | unsigned long sendNTPpacket(IPAddress& address) { |
| 735 | // set all bytes in the buffer to 0 |
| 736 | memset(packetBuffer, 0, NTP\_PACKET\_SIZE); |
| 737 | // Initialize values needed to form NTP request |
| 738 | // (see URL above for details on the packets) |
| 739 | packetBuffer[0] = 0b11100011; // LI, Version, Mode |
| 740 | packetBuffer[1] = 0; // Stratum, or type of clock |
| 741 | packetBuffer[2] = 6; // Polling Interval |
| 742 | packetBuffer[3] = 0xEC; // Peer Clock Precision |
| 743 | // 8 bytes of zero for Root Delay & Root Dispersion |
| 744 | packetBuffer[12] = 49; |
| 745 | packetBuffer[13] = 0x4E; |
| 746 | packetBuffer[14] = 49; |
| 747 | packetBuffer[15] = 52; |
| 748 |  |
| 749 | // all NTP fields have been given values, now |
| 750 | // you can send a packet requesting a timestamp: |
| 751 | udp.beginPacket(address, 123); //NTP requests are to port 123 |
| 752 | udp.write(packetBuffer, NTP\_PACKET\_SIZE); |
| 753 | udp.endPacket(); |
| 754 | } |