

```

1  //////////////////////////////////////
2  //CODI PROGRAMA WASPMOTE PAX////////////////////////////////////
3  //Created By Rubén Almansa for EXIT Research group////////////////////////////////
4  //////////////////////////////////////
5  #include <WaspXBeeZB.h>
6  #include <WaspFrame.h>
7  #include <WaspFrameConstants.h>
8  #define Se 6 //pin sensor
9  #define Sd 8 // pin sensor
10 #define polsador 10
11 #define indicador 13
12 // Adreça Meshlium////////////////////////////////
13 char RX_ADDRESS[] = "0013A200408134BA";
14 //////////////////////////////////
15
16 //DEFINICIÓ VARIABLES COMUNS////////////////////////////////
17 uint8_t error,error2;
18 char* macHigh=" ";
19 char* macLow=" ";
20 timestamp_t time; // guarda el timestsmp del Meshlium
21 unsigned_long epoch; // fa un break del timestamp i ho guarda en aquesta variable
22 int Tx; // registre el flag del frame enviat.
23 int noSend=0; //comptador d'enviaments erronis
24 int errorRTC=0; //registre el flag de la RTC ok.
25 int SumErrorRTC=0; // comptador d'errors de la sincronització deRTC
26 //////////////////////////////////
27
28 //VARIABLES PAX////////////////////////////////
29 int e=0,d=0,tim=0; // entrades i timers a 0
30 int sum =0,res=0; // suma o resta
31 int tot=0; // total de persones
32 char i; //estat de la maquina d'estats
33 int a=1025; // registre eeprom
34 int ad=0; // index de l'adreça eeprom
35 int b=0;
36 int medPax; //calcul per la mitjana persones
37 int c=0;
38 int f;
39 float medPaxSum,deco; // registres interns per calcular les pax
40 int medPaxSumi; // mitjana total que s'envia
41 int bat; // bateria
42 int LDR=0;
43 int RST=0;
44 int reset=0;
45
46 //////////////////////////////////
47 //PROGRAMA D'INICI////////////////////////////////
48 void setup()
49 {
50 /*ACTIVA la RTC i el port USB*/
51
52 RTC.ON();
53 delay(2000);
54 USB.ON();
55 delay(200);
56
57
58 PWR.setSensorPower(SENS_5V,SENS_ON); // dona tensió als pins per alimentar la placa
59 pinMode(DIGITAL6,INPUT);
60 pinMode(DIGITAL8,INPUT);
61 PWR.setWatchdog(WTD_ON,WTD_1S); // configura el watchdog a un segon
62 i='Q';
63 USB.println("START");
64
65 IniXBee();
66 checkNetworkParams();
67 ResetNetwork();
68 RTCMeshlium();
69
70 }
71
72 //////////////////////////////////
73 //PROGRAMA PRINCIPAL////////////////////////////////

```

```

74
75 void loop()
76 {
77
78     USB.println(RTC.getTime());
79     epoch=RTC.getEpochTime(); // guarda el time en un format, és necessari per la
    comanda següent
80     RTC.breakTimeAbsolute( epoch, &time); // trenca el timestamp per obtenir el minut
    que estem
81
82     /* llegeix les entrades constanmet dels sensors IR*/
83     d=digitalRead(DIGITAL8);
84     delay(10);
85
86     e=digitalRead(DIGITAL6);
87     delay(10);
88
89     /* Aquest if serveix per activar el temps en cas que una persona intenta sortir però
    no ho fa*/
90     if (reset==1 && d==LOW && e==LOW) {PWR.setWatchdog(WTD_ON,WTD_1S);}
91
92     if (intFlag & WTD_INT && reset>10)
93     {
94         reset=0;
95         USB.println("WDT");
96         USB.println(tot);
97         intFlag &= ~(WTD_INT);
98         tim=1; // vol dir que el timer WTD s'ha activat
99     }
100
101     ///AQUI COMENÇA LA MAQUINA D'ESTATS PER COMPTAR PERSONES//////////
102
103     // estat de repós o inicial
104     switch (i) {
105         case 'Q': //USB.println("Q");
106             tim=0;
107             if(d==LOW && e==LOW || d==HIGH && e==HIGH ) {i='Q';}
108             if(e==HIGH&&d==LOW) {i='W';} //entrar
109             if(d==HIGH&&e==LOW) {i='T';} // sortir
110
111             break;
112
113
114     // estat on hi ha detecció sensor entrada de la primera persona
115     case 'W':
116         USB.println("w");
117         if(d==LOW && e==LOW ) {reset=reset+1;} // si algú ha però al final no
        entra, el reset relentitza el temps per tornar a 0
118         if(d==HIGH && e==LOW) {i='E';}
119         if(d==LOW && e==HIGH) {i='W';reset=0;intFlag &= ~(WTD_INT);} // passa al
        següent sensor
120         if(d==HIGH && e==HIGH) {i='E';}
121         if(tim==1) {i='Q';tim=0;} // si WTD s'ha activat torna a la pos Ini
122
123         break;
124     // detecta el segon sensor IR
125     case 'E':
126         reset=0;
127         intFlag &= ~(WTD_INT);
128         USB.println("E");
129         if(d==LOW && e==HIGH ) {i='W';}
130         if(d==LOW && e==LOW ) {i='R';}
131         if(d==HIGH && e==LOW) {i='E';}
132         if(d==HIGH && e==HIGH) {i='E';}
133         break;
134
135     // ha fet la seqüència d'entrada i comptabilitza una persona
136     case 'R':
137         //USB.println("R");
138         sum=sum+1; // suma una persona
139         i='Q';
140         tim=0;
141         tot=sum-res; // calcula numero total de persones

```

```

142     USB.println(tot);
143     Utils.writeEEPROM(a,tot);
144     a=a+1; // suma una posició de eeprom
145     b=b+1; // compta el numero de deteccions completes
146
147     break;
148
149 // es el mateix procés que l'anterior però per sortir
150 case 'T':
151     USB.println("T");
152     if(d==LOW && e==LOW){reset=reset+1;}
153     if(d==LOW && e==HIGH){i='Y';}
154     if(d==HIGH && e==LOW){i='T';reset=0;intFlag &= ~(WTD_INT);}
155     if(d==HIGH && e==HIGH){i='Y';}
156     if(tim==1){i='Q';tim=0;}
157
158     break;
159
160 case 'Y':
161     reset=0;
162     intFlag &= ~(WTD_INT);
163     USB.println("Y");
164     if(d==HIGH && e==LOW){i='T';}
165     if(d==LOW && e==LOW){i='U';}
166     if(d==LOW && e==HIGH){i='Y';}
167     if(d==HIGH && e==HIGH){i='Y';}
168     break;
169
170 case 'U':
171     //USB.println("U");
172     res=res+1;
173     i='Q';
174     tot=sum-res;
175     Utils.writeEEPROM(a,tot);
176     a=a+1;
177     b=b+1;
178     USB.println(tot);
179
180     break;}
181
182 //Comprovem la hora amb aquest if////////
183
184 if( ((time.minute == 00) || (time.minute == 10) || (time.minute == 20) || (time.minute
== 30) || (time.minute == 40) || (time.minute == 50)) && Tx==0)
185
186 {
187     bat=PWR.getBatteryLevel();
188     ad=1025;/////s'ha de canviar
189     f=1;
190     medPax=0;
191     // suma tots els registres de la eeprom per fer la mitjana
192     while(f<=b){
193         medPax=(Utils.readEEPROM(ad));
194         USB.println(medPax);
195         medPaxSum=medPax+medPaxSum;
196         ad=ad+1;
197         f=f+1;
198     }
199     // fa la mitjana
200     medPaxSum=medPaxSum/b;
201     medPaxSumi=medPaxSum;
202     deco=medPaxSum-medPaxSumi;
203     if(deco>0.5){medPaxSumi=medPaxSumi+1;} // fem aproximació de gent
204     USB.println(medPaxSum);
205     USB.println(medPaxSumi);
206     //if(medPaxSumi<0){medPaxSumi=0;}
207     //if(medPaxSumi>14){medPaxSumi=14;}
208     medPaxSum=0;
209     LDR=analogRead(ANALOG3); // llegeix el valor per saber si es fosc i resetjar el
valor pax
210     LDR=(LDR*5)/1024;
211     USB.println(LDR);
212     delay(100);

```

```

213     a=1025; // a és adreça de EEPROM
214     b=1; // b és el número de vegades que ha entrat o sortit persones b=1 quan hi ha
        gent b=0 quan ha marxat tothom
215     if(tot<=0){tot=0;medPaxSumi=0;}
216     Utils.writeEEPROM(a,tot);
217     a=a+1;
218     sum=tot; // la suma passa a ser el total
219     res=0;
220     EnviarDades(); // envia les dades
221     USB.println(LDR);
222     if(LDR>1){RST=0;} // guarda variable RST per resetejar el sistema una vegada
223     // el ssistema torna a 0 pax
224     if((LDR<1 && RST==0)&& (time.hour == 01)){sum = 0; res=0; tot=0; a=1025; b=0;
        RST=1;PWR.reboot();}
225     // alarma per resetejar la variable de TX passat una estona
226     RTC.setAlarm1("00:00:08:00",RTC_OFFSET,RTC_ALM1_MODE4);
227 }
228 if ( intFlag & RTC_INT ) {
229     intFlag &= ~(RTC_INT);
230     RTC.clearAlarmFlag();
231     delay(1000);
232     Tx=0;
233 }
234
235 }
236
237
238 ///////////////////////////////////////////////////
239 //*****AQUESTA FUNCIO S'ENCARREGA D'ENVIAR LES DADES*//////////
240
241 void EnviarDades() {
242
243     while(Tx==0){
244
245         frame.setID( macLow ); // és el identificador del WASPMOTE
246         frame.createFrame(ASCII); // indicador per iniciar el frame.
247         //s'afegeix les variables de cada waspmote que s'enviaràn
248         frame.addSensor(SENSOR_STR, medPaxSumi,tot,bat);
249         delay(100);
250         // envia el paquet
251         error = xbeeZB.send( RX_ADDRESS, frame.buffer, frame.length );
252         if( error == 0 ) // comprova l'enviament
253
254         {
255             USB.println(F("send ok"));
256
257             // blink green LED
258             Utils.blinkGreenLED();
259             noSend=0;
260             Tx=1; // Variable per confirmar qu l'envio ha estat correcte
261             delay(60000);
262         }
263         else
264         {
265             USB.println(F("send error"));
266
267             if (noSend>10){PWR.reboot();}
268             noSend=noSend+1;
269         }
270     }
271 }
272
273 ///////////////////////////////////////////////////
274
275 //***** FUNCIO QUE CONNECTA AMB MESHLIUM PER OBTENIR EL TIMESTAMP*//////////
276
277 void RTCMeshlium()
278 {
279     while (errorRTC ==0){
280         error2 = xbeeZB.setRTCfromMeshlium(RX_ADDRESS);
281         // check flag si la RTC s'ha sincronitzat
282         if( error2 == 0 )

```

```

284 {
285     USB.print(F("SET RTC ok. "));
286     errorRTC=1;
287     USB.println(RTC.getTime());
288 }
289 else
290 {
291     USB.print(F("SET RTC error. "));
292     errorRTC=0;
293     SumErrorRTC=SumErrorRTC+1;
294     if (SumErrorRTC>10){PWR.reboot();} // si es produeixen molts errors reseteja el
        sistema.
295     }
296 }
297 }
298 ///////////////////////////////////////////////////
299 //*****FUNCIÓ ENCARREGADA D'INICIAR ELS XBEE I OBTENIR LA MAC*/////////
300
301 void IniXBee()
302 {
303 {
304     xbeeZB.ON();
305     delay(3000);
306     xbeeZB.getOwnMacLow(); // obté la MAC LOW
307     xbeeZB.getOwnMacHigh(); // obté la MAC HIGH
308     // Guarda les MAC
309     Utils.hex2str(xbeeZB.sourceMacHigh,macHigh,4);
310     Utils.hex2str(xbeeZB.sourceMacLow,macLow,4);
311 }
312
313 ///////////////////////////////////////////////////
314 //*****COMPROVA ELS PARAMETRES DE LA XARXA XBEE*/////////
315 void checkNetworkParams()
316 {
317     // Obté la identificació de xarxa
318     xbeeZB.getOperating64PAN();
319
320     // comprova la associació amb meshlium
321     xbeeZB.getAssociationIndication();
322
323     while( xbeeZB.associationIndication != 0 )
324     {
325         delay(2000);
326
327         xbeeZB.getOperating64PAN();
328
329         USB.print(F("operating 64-b PAN ID: "));
330         USB.printHex(xbeeZB.operating64PAN[0]);
331         USB.printHex(xbeeZB.operating64PAN[1]);
332         USB.printHex(xbeeZB.operating64PAN[2]);
333         USB.printHex(xbeeZB.operating64PAN[3]);
334         USB.printHex(xbeeZB.operating64PAN[4]);
335         USB.printHex(xbeeZB.operating64PAN[5]);
336         USB.printHex(xbeeZB.operating64PAN[6]);
337         USB.printHex(xbeeZB.operating64PAN[7]);
338         USB.println();
339
340         xbeeZB.getAssociationIndication();
341     }
342
343     USB.println(F("\nJoined a network!"));
344
345     xbeeZB.getOperating16PAN();
346     xbeeZB.getOperating64PAN();
347     xbeeZB.getChannel();
348
349     USB.print(F("operating 16-b PAN ID: "));
350     USB.printHex(xbeeZB.operating16PAN[0]);
351     USB.printHex(xbeeZB.operating16PAN[1]);
352     USB.println();
353
354
355

```

```

356     USB.print(F("operating 64-b PAN ID: "));
357     USB.printHex(xbeeZB.operating64PAN[0]);
358     USB.printHex(xbeeZB.operating64PAN[1]);
359     USB.printHex(xbeeZB.operating64PAN[2]);
360     USB.printHex(xbeeZB.operating64PAN[3]);
361     USB.printHex(xbeeZB.operating64PAN[4]);
362     USB.printHex(xbeeZB.operating64PAN[5]);
363     USB.printHex(xbeeZB.operating64PAN[6]);
364     USB.printHex(xbeeZB.operating64PAN[7]);
365     USB.println();
366
367     USB.print(F("channel: "));
368     USB.printHex(xbeeZB.channel);
369     USB.println();
370
371 }
372 ///////////////////////////////////////////////////
373 //*****RESETEJA ELS PAREMETRES DE LA XARXA*/////////
374
375 void ResetNetwork(){
376
377     while((xbeeZB.channel) !=0x10){
378         xbeeZB.OFF();
379         delay(1000);
380         xbeeZB.ON();
381         delay(3000);
382         xbeeZB.resetNetwork(0); // reseteja els paràmetres de la xarxa
383         delay(1000);
384         checkNetworkParams(); // comprova aquests paràmetres.
385     }
386 }
387
388

```