Sistemas Digitales
Trabajo Final

Martín Noblía

Profesores:

Jose Juarez
Eric Pernia

Universidad Nacional de Quilmes

2016

1 Introduccion

```
void prvAnemometerTaks(void *pvParameters)
{
   /* Anemometer pin states */
   typedef enum{PIN_UP, PIN_FALLING, PIN_DOWN, PIN_RISING} pin_state_t;
   /* Auxiliar variables */
   portBASE_TYPE xFreq = 0;
   portBASE_TYPE xCounter = 0;
   portBASE_TYPE xTemp = 0;
   /* initial condition */
   pin_state_t pin_state = PIN_UP;
   /* message data */
   xMetaData xAnemometerMessage;
   /* message flag for the Gatekeeper */
   xAnemometerMessage.xSource = SENDER_ANEMOMETER;
   /* Task processig */
   while(1)
   {

      /* MEF for counting the states changes */
      switch(pin_state)
      {
         case PIN_UP:
            {
               if(!digitalRead(DIO32))
               {
                  pin_state = PIN_FALLING;
               }
               break;
            }
         case PIN_FALLING:
            {
               xCounter += 3;
               if(!digitalRead(DIO32))
               {
                  pin_state = PIN_DOWN;
                  xFreq++;
                  digitalWrite(LEDR, ON);
               }
               else
                  pin_state = PIN_UP;
               break;
            }
         case PIN_DOWN:
               {
               if(digitalRead(DIO32))
               {
                  pin_state = PIN_RISING;
               }
               break;
```

```
            }
         case PIN_RISING:
            {
               xCounter += 3;
               if(digitalRead(DIO32))
               {
                  pin_state = PIN_UP;
                  digitalWrite(LEDR, OFF);
               }
               else
               {
                  pin_state = PIN_DOWN;
               }
               break;
             }
         }

      if(xSemaphoreTake(xTimeSignal, ( TickType_t )0))
      {
         if(xFreq > FREQUENCY_ALARM_THRESHOLD_1)
         {
            xTemp = ALARM_MESSAGE_1;
            xQueueSendToBack(xALARMQueue, (void *)&xTemp, portMAX_DELAY);
         }
         if(xFreq > FREQUENCY_ALARM_THRESHOLD_2)
         {
            xTemp = ALARM_MESSAGE_2;
            xQueueSendToBack(xALARMQueue, (void *)&xTemp, portMAX_DELAY);
         }
         /* The time message arrive --> prepare the message package */
         xAnemometerMessage.xMessage = xFreq;
         /* send the package via the Gatekeeper */
         xQueueSendToBack(xUARTQueue, (void *)&xAnemometerMessage, ( TickType_t )0);
         /* reset the values */
         xFreq = 0;
         xCounter = 0;
      }
   }
```

Referencias

[1] Digital communications, fundamentals and applications. Bernard Sklar

[2] Julia: A fresh approach to numerical computing. Jeff Bezanson, Alan Edelman, Stefan Karpinski, Viral B. Shah(2014)`http://arxiv.org/abs/1411.1607`