```
//ESTE PROGRAMA GESTIONA UN IMU DE 6 DOF DE SALIDAS DIGITALES COMPATIBLE CON EL PROTOCOLO I2C

void main()
{
// DECLARACION DE LOS REGISTROS DEL ACELEROMETRO
 #define READ_ADXL    0xA7
 #define WRITE_ADXL   0xA6
 #define DEVID        0x00
 #define OFSX         OxIE
 #define OFSY         OxIF
 #define OFSZ         Ox20
 #define WINDOW       0x23
 #define DATA_FORMAT  0x31
 #define POWERCTL     0x2D
 #define INT_ENABLE   0x2E
 #define BWRATE       0x2C
 #define DATAX0       0x32
 #define DATAX1       0x33
 #define DATAY0       0x34
 #define DATAY1       0x35
 #define DATAZ0       0x36
 #define DATAZ1       0x37

 // DECLARACION DE LOS REGISTROS DEL GIROSCOPO

 #define READ_ITG     0xD3
 #define WRITE_ITG    0xD2
 #define SAMPLE_DIV   0x15
 #define DLPF_FS      0x16
 #define TEMP_HI      0x1B
 #define TEMP_LO      0x1C
 #define GYRO_X1      0x1D
 #define GYRO_X0      0x1E
 #define GYRO_Y1      0x1F
 #define GYRO_Y0      0x20
 #define GYRO_Z1      0x21
 #define GYRO_Z0      0x22
.

int k,A1,B1,A2,B2;
signed int C2,C1;
char txt1[7],txt2[7];
// CONFIGURACION DEL PUERTO B Y C COMO SALIDA

 DDRC =  1;
 DDRB =  1;
 PORTC = 1;
 PORTB = 2;

 TWI_Init(400000);         // initialize TWI communication

  //---------------------  INICIACLIZACION DEL ACELEROMETRO  -------------------

   TWI_Start();              // issue TWI start signal
   TWI_Write(WRITE_ADXL);    // send byte via TWI (device address + W)
   TWI_Write(DATA_FORMAT);
   TWI_Write(0X08);
   TWI_Stop();               // issue TWI stop signal
   Delay_ms(1);

   TWI_Start();              // issue TWI start signal
   TWI_Write(WRITE_ADXL);    // send byte via TWI (device address + W)
   TWI_Write(BWRATE);
   TWI_Write(0X0F);
   TWI_Stop();               // issue TWI stop signal
   Delay_ms(1);

   TWI_Start();              // issue TWI start signal
   TWI_Write(WRITE_ADXL);    // send byte via TWI (device address + W)
   TWI_Write(POWERCTL);
   TWI_Write(0X08);
   TWI_Stop();               // issue TWI stop signal
   Delay_ms(1);

   TWI_Start();              // issue TWI start signal
   TWI_Write(WRITE_ADXL);    // send byte via TWI (device address + W)
   TWI_Write(INT_ENABLE);
   TWI_Write(0X80);
   TWI_Stop();                 // issue TWI stop signal
   Delay_ms(1);

  //---------------------------O-------------------------------------------------//

   TWI_Start();                // INICIALIZACION DEL GYROSCOPIO
   TWI_Write(WRITE_ITG);
   TWI_Write(DLPF_FS);
   TWI_Write(0X18);
   TWI_Stop();
   Delay_ms(1);

   TWI_Start();                // INICIALIZACION DEL GYROSCOPIO
   TWI_Write(WRITE_ITG);
   TWI_Write(SAMPLE_DIV);
   TWI_Write(0X00);
   TWI_Stop();
   Delay_ms(1);

 UART1_Init(115200);       // Initialize UART module at 11520000 bps
 Delay_ms(100);
 while(1)
  {
     if(UART1_Data_Ready()==1)

     {
         if(UART1_Read() == '@')

  //----------------------  DATOS DEL ACELEROMETRO  --------------------------------
             {
              TWI_Start();                 // issue TWI start signal
              TWI_Write(WRITE_ADXL);       // send byte via TWI (device address + W)
              TWI_Write(DATAX0);
              TWI_Start();
              TWI_Write(READ_ADXL);

                    for(k=0;k<3;k++)
                         {
                          A1=TWI_Read(1);
                          B1=TWI_Read(1);
                          //UART1_Write(B1);
                          //UART1_Write(A1);
                          C1=B1*0x100+A1;
                          IntToStr(C1, txt1);
                          UART1_Write_Text(txt1);
                          //UART1_Write(32);
                          A1=B1=C1=0;
                          Delay_ms(1);
                         }
              TWI_Stop();
              Delay_ms(1);

  //----------------------  DATOS DEL GYROSCOPIO  --------------------------------
              TWI_Start();               // issue TWI start signal
              TWI_Write(WRITE_ITG);      // send byte via TWI (device address + W)
              TWI_Write(GYRO_X1);
              TWI_Start();                    // issue TWI start signal
              TWI_Write(READ_ITG);          // send byte via TWI (device address + W)

                     for(k=0;k<3;k++)
                         {
                          A2=TWI_Read(1);
                          B2=TWI_Read(1);
                          //UART1_Write(A2);
                          //UART1_Write(B2);
                          C2=A2*0x100+B2;
                          IntToStr(C2, txt2);
                          UART1_Write_Text(txt2);
                          //UART1_Write(32);   //espacio ascii
                          A2=B2=C2=0;
                          Delay_ms(1);
                         }
            TWI_Stop();
            TWI_Close();               // issue TWI stop signal
           }
        }
     }
}
```