



*Aguijón*

---

*Notas de aplicación*

## **Nota de aplicación 36:**

### **Matrix Keyboard**

#### **Descripción:**

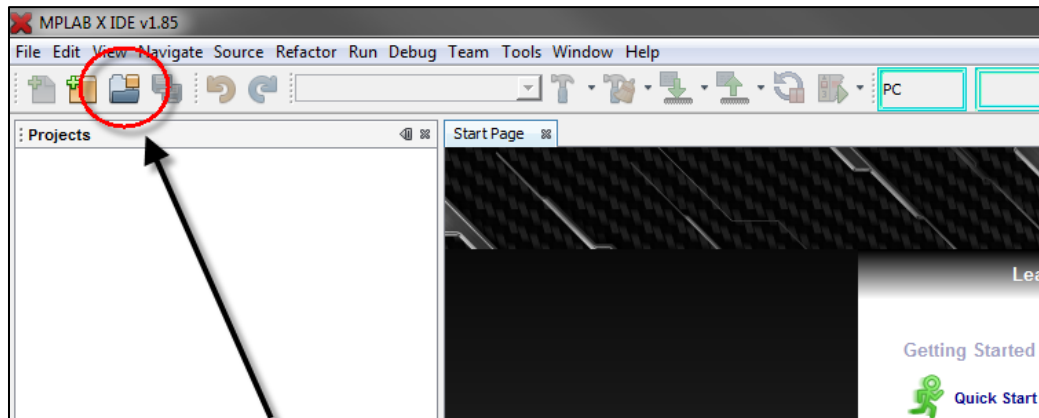
Utilizar un teclado matricial y conocer las declaraciones necesarias para su correcto funcionamiento

#### **Herramientas:**

1. Aguijón 3.0, Aguijón 4.0 ó Aguijón 4.1
2. MPLAB X®
3. Aguijón HID bootloader
4. Cable USB 'A' to 'A'
5. Librerías HammerHead.
6. Teclado Matricial

## Pasos:

1. Abrir MPLAB X® y cargar el archivo del proyecto: **Application Note 36.X**



Haz 'clic' aquí y  
abre el proyecto

2. Abrir el archivo **main.c**



### 3. Ir a la línea #33

```
26 | *****/
27 | #include "aguijon.h"
28 | #include "EEPROM.h"           //EEPROM support
29 | #include "IO.h"              //IO support
30 | #include "inputs.h"          //inputs support
31 | #include "LCD.h"             //LCD support
32 | #include "outputs.h"         //output support
33 | #include "matrix_keyboard.h" //matrix keyboard support
34 |
35 |
36 |
37 |
38 |
39 |
40 | //////////////////////////////////
```

- Se ha agregado una nueva cabecera (Header). En esta librería se encuentran las definiciones que se deben llevar acabo dependiendo de la configuración de cual puerto el usuario desea configurar. Se manejaran para un teclado matricial de 5 x 5, por lo que cuenta con 5 columnas y 5 líneas. Las columnas se han declarado como salidas y las 5 líneas como entradas. Si el teclado matricial que se desea conectar no es del rango del rango mencionado, solo se tiene que conectar las columnas y las líneas que el teclado con las que el teclado matricial cuenta. Se explicara más a detalle más adelante.
- Se puede configurar los puerto J5 y J6. Se podrán seleccionar ya sea el J5 o el J6 o ambos. Las configuraciones de entrada y salida (I/o) se pueden observar en el archivo "IO.c".
- Las definiciones para seleccionar cual puerto se va a utilizar serán definidas en el archivo "matrix\_keyboard.h".

4. El código está programado para utilizar el puerto J5, si desea cambiar, ir a la línea #29

```
26 | *****/
27 | #include "aguijon.h"
28 | #include "EEPROM.h"           //EEPROM support
29 | #include "IO.h"              //IO support
30 | #include "inputs.h"          //inputs support
31 | #include "LCD.h"             //LCD support
32 | #include "outputs.h"         //output support
33 | #include "matrix keyboard.h" //matrix keyboard support
34 |
35 |
36 |
37 |
38 |
39 |
```

- Si se va a cambiar la configuración inicial, se presionara la tetra CTRL y apuntaremos con el puntero a el nombre del archivo y nos mandara a dicho archivo.
- Si no se desea cambiar la configuración saltar el paso 4 y 5.

5. Dentro del archivo “matrix\_keyboard.h” iremos a la línea #42

```
37  * you want to use.
38  *
39  * You can select both ports */
40
41
42  #define MATRIX_KEYBOARD_PORT1_J5    //<----- PORT 1 J5
43
44  #define MATRIX_KEYBOARD_PORT2_J6    //<----- PORT 2 J6
45
46  /*****/
47
48
```

- Se puede observar dos definiciones, una habilitada y otra comentada. Cada una habilita un puerto de expansión para utilizarse como entradas y salidas para el teclado matricial, si se desea cambiar dicha configuración, se puede quitar el comentario de la línea 44 para que los dos estén habilitados o, habilitar la línea 44 y comentar la línea 42, para que solo el puerto J6 esté habilitado.

6. Si ha sido configurado correctamente el puerto o los puertos que se van a utilizar, ir a línea #76

```
69 unsigned char key = 0;
70 unsigned char key2 = 0;
71 char lcdMSG[20];
72
73
74 HammerHead_Init(); //initialize [VD]HammerHead
75 LCD_IntroAnimation();
76 LCD_PutStr(1,0,"Vinagron Digital ",ON);
77 LCD_PutStr(2,0,"Application Note 36",OFF);
78 delayms(1000);
79
80 //Create a String for LCD_PutStr
81 LCD_PutStr(1,0,"Matrix Keyboard ",ON);
82 delayms(1000);
83
```

**LCD\_PutStr (int y, int x, char \*msg, BOOL clear);**

- Función que muestra una cadena de caracteres en la pantalla LCD; donde:  
**Int y** = Coordenada en y (Valor entero del 1 al 2.)  
**Int X** = Coordenada en x (Valor entero del 0 al 20.)  
**Char \*msg** = Cadena de caracteres (De 0 a 20 caracteres)  
**BOOL clear** = Determina si se borra la pantalla antes de escribir  
(TRUE =Borrar, FALSE =No borrar).

## 7. Ir a línea #78

```
75 LCD_IntroAnimation();
76 LCD_PutStr(1,0,"Vinagron Digital ",ON);
77 LCD_PutStr(2,0,"Application Note 36",OFF);
78 delayms(1000);
79
80 //Create a String for LCD_PutStr
81 LCD_PutStr(1,0,"Matrix Keyboard ",ON);
82 delayms(1000);
83
84
85
86
87 for(;;){
88
89
```

### **Delayms (ms);**

- Función que Pausa el programa por un tiempo determinado (en milisegundos); donde:  
**ms** = el número de milisegundos que se desea pausar el programa.



8. Ir a la línea #90

```
83
84
85
86
87     for(;;){
88
89
90         key = Matrix_Keyboard_Read_J5();
91         key2 = Matrix_Keyboard_ReadPort_J5();
92
93
94         //Create a String for LCD_PutStr
95         sprintf(lcdMSG,"Value: %c ",key2);
96         LCD_PutStr(1,0,"Toggle the RLY'S ",OFF);
97         LCD_PutStr(2,0,lcdMSG,OFF);
```

**Matrix\_Keyboard\_Read\_J5 ( );**

- Esta función retornara el valor leído por el teclado matricial. **Regresara solo una vez el valor leído.**
- Retornara un valor de tipo “unsigned char”, el valor varía dependiendo del teclado matricial que se esté utilizando.
- La configuración de los valores que regresa el teclado matricial 5x5 son de la siguiente manera:

|         |      |      |      |      |      |
|---------|------|------|------|------|------|
| Línea 5 | Null | Null | Null | Null | Null |
| Línea 4 | 1    | 2    | 3    | A    | Null |
| Línea 3 | 4    | 5    | 6    | B    | Null |
| Línea 2 | 7    | 8    | 9    | C    | Null |
| Línea 1 | *    | 0    | #    | D    | Null |

|  |           |           |           |           |           |
|--|-----------|-----------|-----------|-----------|-----------|
|  | Columna 1 | Columna 2 | Columna 3 | Columna 4 | Columna 5 |
|--|-----------|-----------|-----------|-----------|-----------|

9. Ir a la línea #94

```

84
85
86
87     for (;;) {
88
89
90         key = Matrix_Keyboard_Read_J5 ();
91         key2 = Matrix_Keyboard_ReadPort_J5 ();
92
93
94         //Create a String for LCD_PutStr
95         sprintf(lcdMSG, "Value: %c ", key2);
96         LCD_PutStr(1, 0, "Toggle the RLY'S ", OFF);
97         LCD_PutStr(2, 0, lcdMSG, OFF);
98

```

**Matrix\_Keyboard\_ReadPort\_J5 ();**

- Esta función retornara el valor leído por el teclado matricial. **Regresara continuamente el valor el teclado matricial este leyendo.**
- Retornara un valor de tipo “unsigned char”, el valor varía dependiendo del teclado matricial que se esté utilizando.

|         |      |      |      |      |      |
|---------|------|------|------|------|------|
| Línea 5 | Null | Null | Null | Null | Null |
| Línea 4 | 1    | 2    | 3    | A    | Null |
| Línea 3 | 4    | 5    | 6    | B    | Null |
| Línea 2 | 7    | 8    | 9    | C    | Null |
| Línea 1 | *    | 0    | #    | D    | Null |

|  |           |           |           |           |           |
|--|-----------|-----------|-----------|-----------|-----------|
|  | Columna 1 | Columna 2 | Columna 3 | Columna 4 | Columna 5 |
|--|-----------|-----------|-----------|-----------|-----------|

10. Ir a línea #95

```

88
89
90     key  = Matrix_Keyboard_Read_J5();
91     key2 = Matrix_Keyboard_ReadPort_J5();
92
93
94     //Create a String for LCD_PutStr
95     sprintf(lcdMSG, "Value: %c ", key2);
96     LCD_PutStr(1, 0, "Toggle the RLY'S ", OFF);
97     LCD_PutStr(2, 0, lcdMSG, OFF);
98
99
100     switch(key) {
101
102         case '1':

```

**sprintf (char \*, const char \*, ...);**

- Función que Genera una cadena de caracteres y lo asigna a una variable; Donde:  
**Char \*** = Variable a asignar cadena de caracteres (Variable de tipo char)  
**const char \***, = Cadena de caracteres a asignar a la variable.

11. Ir a línea #103

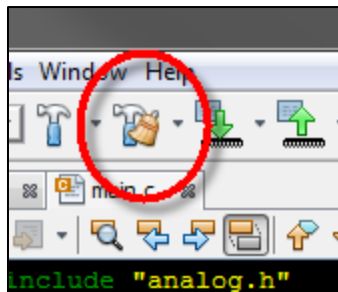
```
96     LCD_PutStr(1,0,"Toggle the RLY'S ",OFF);
97     LCD_PutStr(2,0,lcdMSG,OFF);
98
99
100     switch(key) {
101
102         case '1':
103             RLY_Toggle(1);
104             break;
105
106         case '2':
107             RLY_Toggle(2);
108             break;
109
110         case '3':
```

***RLY\_Toggle (int num);***

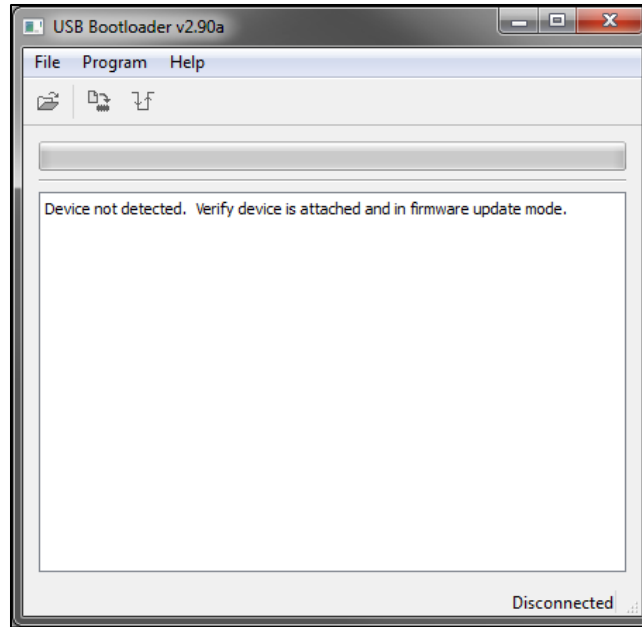
- *Función que cambia el estado lógico de un relevador cada vez que sea llamada donde:*
- ***Int num*** = *Sera el relevador que cambiara de estado lógico.*

## 12. Compilar y programar

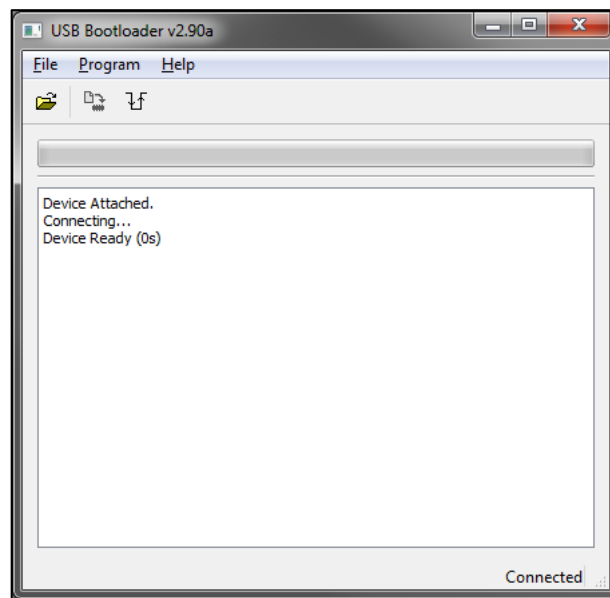
Al hacer clic en el ícono de compilar, y si no hay errores de compilación, el bootloader será cargado automáticamente.



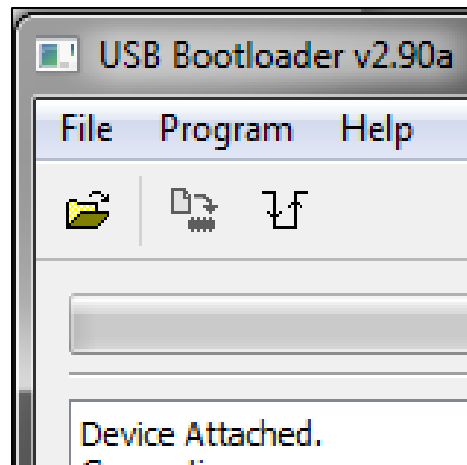
Cuando aparezca la ventana del bootloader, presione el Push-Button número 1 y conecte la fuente de voltaje o encienda el Aguijón y mantenga el PB1 presionado hasta que los LEDs empiecen a parpadear.



*La ventana del Bootloader indicará la conexión establecida con el aguijón:*

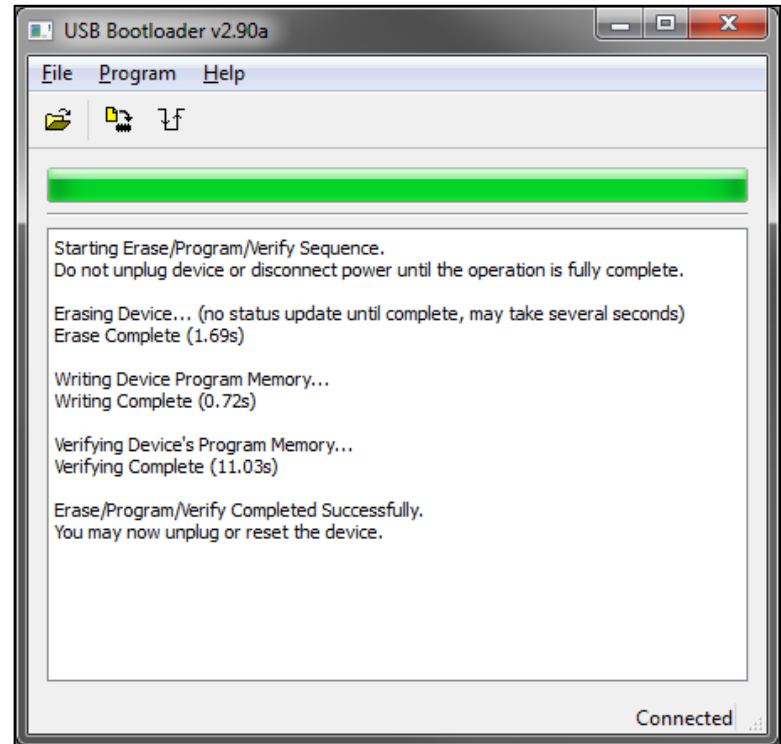
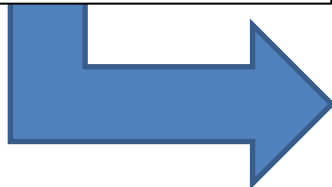
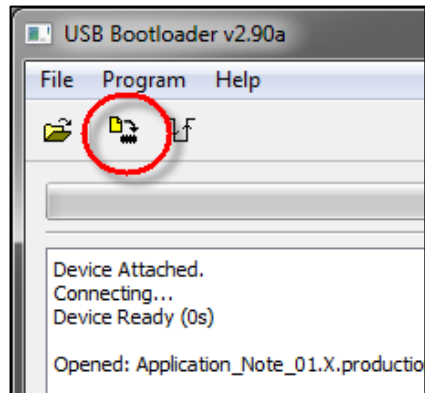


13. Hacer Clic en Abrir y Cargar el archivo **Application Note 31.X.production.hex**

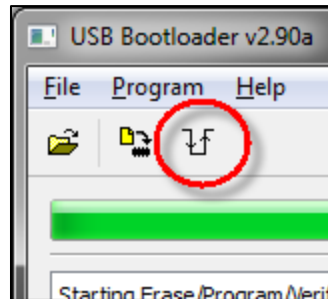


*El archivo, depende de la plataforma de hardware.*

*Una vez cargado el archivo, hacer clic en el ícono de programar.*



Hacer clic en el ícono de Reset cuando en la ventana del Bootloader se indique que se terminó de programar con éxito.



Una vez programado podemos verificar el programa corriendo en la tarjeta.

14. Para verificar el funcionamiento del programa presionar los botones de del número 1 al 4 y verificar que los relevadores del agujón cambien de estado lógico.