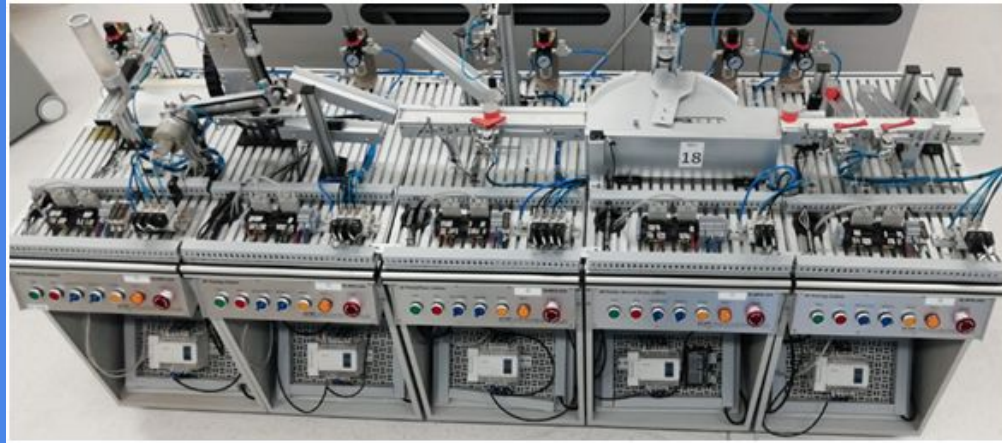


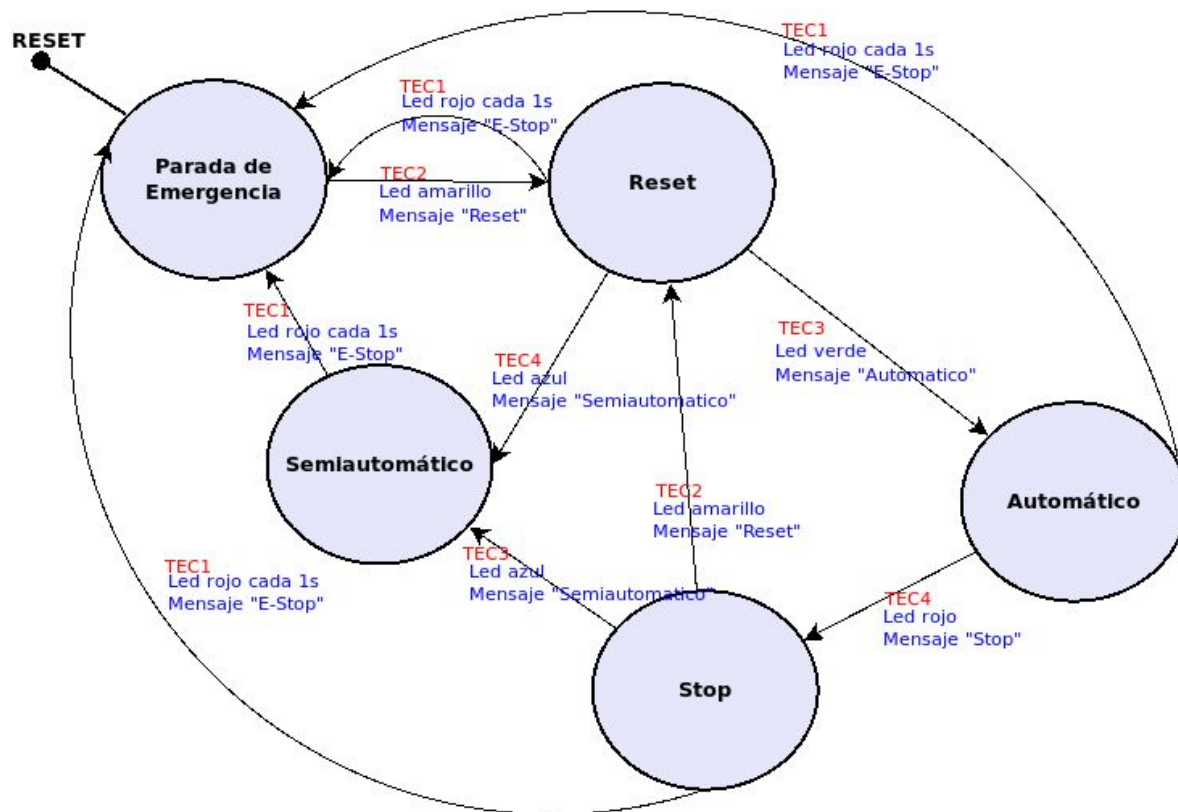
## Control y visualización de los modos de funcionamiento de un Sistema Modular Flexible



**Autor : Fernando Tajés**

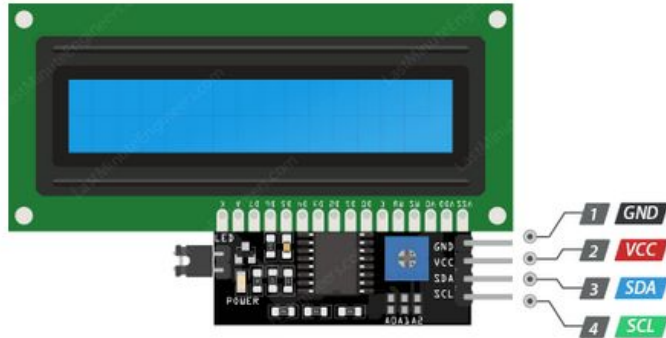
# Máquina de estados

- Máquina de estados de la solución implementada



# Display LCD y expensor PCF8574

- Display LCD 16x2
- PCF8574 Remote 8-Bit I/O Expander for I2C Bus
- EDU-CIAA-NXP

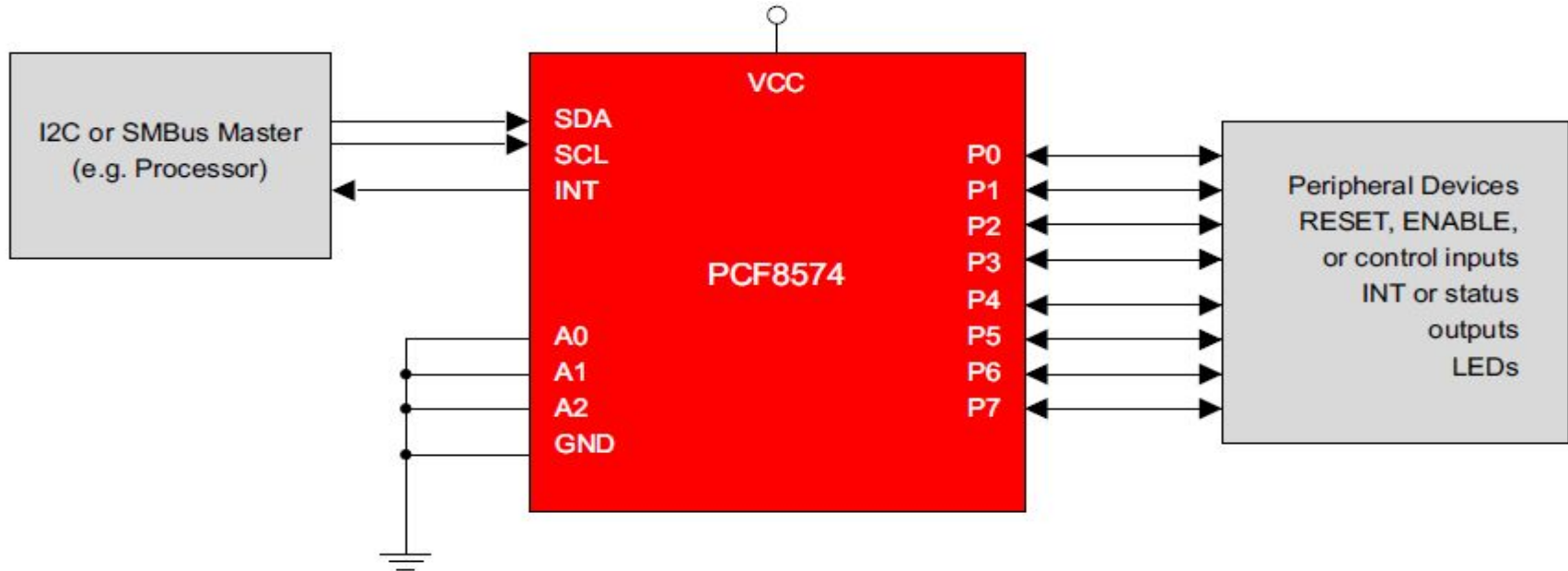


<http://www.proyecto-ciaa.com.ar/devwiki/doku.php?id=desarrollo:edu-ciaa:edu-ciaa-nxp>

Extraída de <https://www.majju.pk/product/pct8574-i2c-display-module-for-lcd/>

# Expansor PCF8574

- PCF8574 Remote 8-Bit I/O Expander for I2C Bus



Extraída del datasheets PCF8574 de Texas Instruments

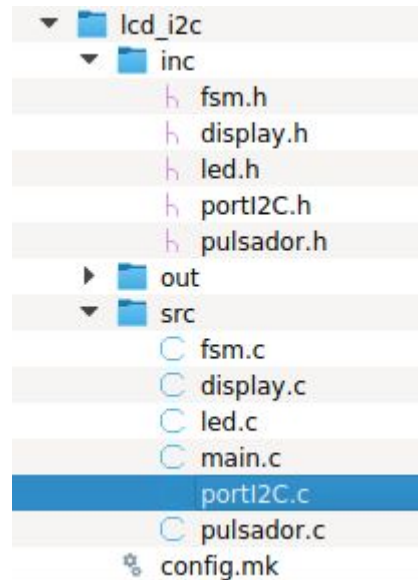
# Display LCD

- Display LCD 16x2 módulo WH1602A-YYH-ET

Pin No.	Symbol	Level	Description
1	V <sub>SS</sub>	0V	Ground
2	V <sub>DD</sub>	5.0V	Supply Voltage for logic
3	VO	(Variable)	Operating voltage for LCD
4	RS	H/L	H: DATA, L: Instruction code
5	R/W	H/L	H: Read(MPU→Module) L: Write(MPU→Module)
6	E	H,H→L	Chip enable signal
7	DB0	H/L	Data bus line
8	DB1	H/L	Data bus line
9	DB2	H/L	Data bus line
10	DB3	H/L	Data bus line
11	DB4	H/L	Data bus line
12	DB5	H/L	Data bus line
13	DB6	H/L	Data bus line
14	DB7	H/L	Data bus line
15	A	—	LED +
16	K	—	LED —

# Software

- Programa



```
portI2C.c
1  #include "portI2C.h"
2
3  #define DEFAULT_I2C I2C0
4  #define SPEED_100KHZ 100000
5
6  void portI2Cinit()
7  {
8      i2cInit( DEFAULT_I2C, SPEED_100KHZ );
9  }
10
11 void portLcdInit( uint16_t rows, uint16_t columns, uint16_t pixelH, uint16_t pixelV )
12 {
13     lcdInit( rows, columns, pixelH, pixelV );
14 }
15
16 void driverGpioWrite(pcf8574T_gpio_t pin, bool_t valor){
17     pcf8574TGpioWrite( pin, valor);
18 }
19
20 void portDelay_us(uint32_t us)
21 {
22     lcdDelay_us(us);
23 }
24
25 void portDelay_ms(uint32_t ms)
26 {
```



# Software

- Programa

```
fsm.c
1  #include "fsm.h"
2  #include "led.h"
3  #include "pulsador.h"
4  #include "sapi.h"
5  #include "funciones.h"
6  #include "portI2C.h"
7
8  #define FILAS 2
9  #define COLUMNAS 16
10 #define PIXEL_H 5
11 #define PIXEL_V 8
12 #define RETARDO 200
13
14 typedef enum { //Modos de funcionamiento de la Planta
15     PARADA_EMERGENCIA,
16     REINICIAR,
17     AUTOMATICO,
18     STOP,
19     SEMIAUTOMATICO
20 } fsmModosFuncionamiento_t;
21
22 static fsmModosFuncionamiento_t fsmModosFuncionamiento;
23 static void fsmModosFuncionamientoError( void );
24
25 void fsmModosFuncionamientoActualizar(void){
26     limpiarPantalla();
```

```
fsm.c
34 while(1) {
35
36     switch( fsmModosFuncionamiento ){
37
38         case PARADA_EMERGENCIA: /* PARADA DE EMERGENCIA */
39             //Parpadeo del LED2 cada 200ms
40             if ( delayRead( &delays ) ){
41                 if( !led )
42                     led = ON;
43                 else
44                     led = OFF;
45                 gpioWrite( LED2, led );
46             } //END IF Parpadeo
47             //REINICIAR
48             if ( estadoPulsador(TEC2)){
49                 ledApagar(LED2);
50                 limpiarPantalla();
51                 printPantalla("RESET");
52                 fsmModosFuncionamiento = REINICIAR;
53             }
54             break;
55
56         case REINICIAR: /* RESET */
57             ledEncender(LED1);
58             //SEMIAUTOMATICO
59             if ( estadoPulsador(TEC4)){
60                 ledApagar(LED1);
61                 limpiarPantalla();
62                 printPantalla("SEMIAUTOMATICO");
63                 fsmModosFuncionamiento = SEMIAUTOMATICO;
64             }
65             //AUTOMATICO
66             if ( !estadoPulsador(TEC2) && estadoPulsador(TEC3)){
67                 ledApagar(LED1);
68                 limpiarPantalla();
69                 printPantalla("AUTOMATICO");
70                 fsmModosFuncionamiento = AUTOMATICO;
```

**GRACIAS POR SU ATENCION**