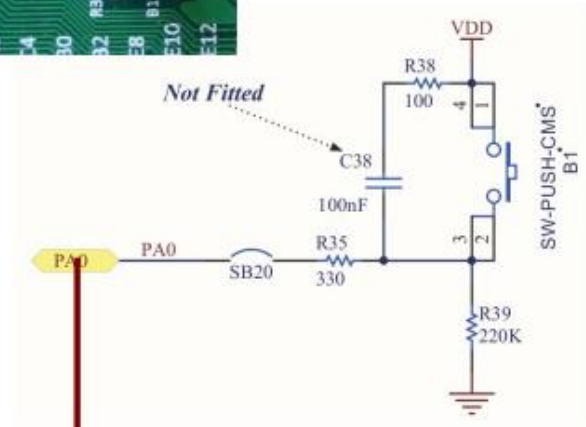
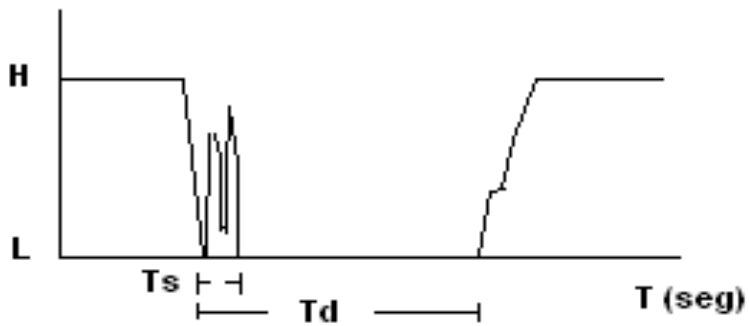
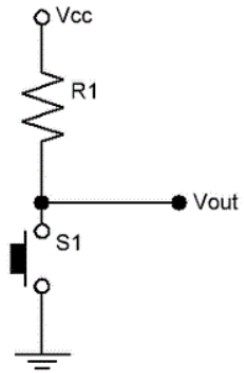
MICROS 32 BITS
STM - GPIO

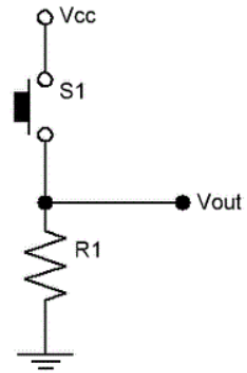
ROBINSON JIMENEZ MORENO



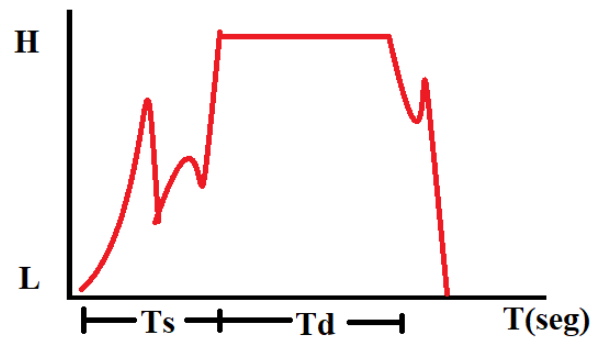




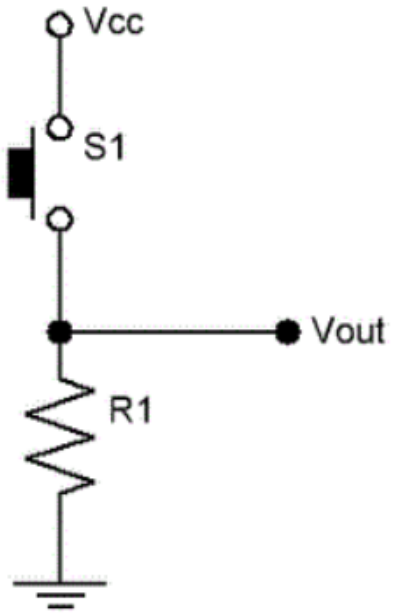
Pull Up



Pull Down

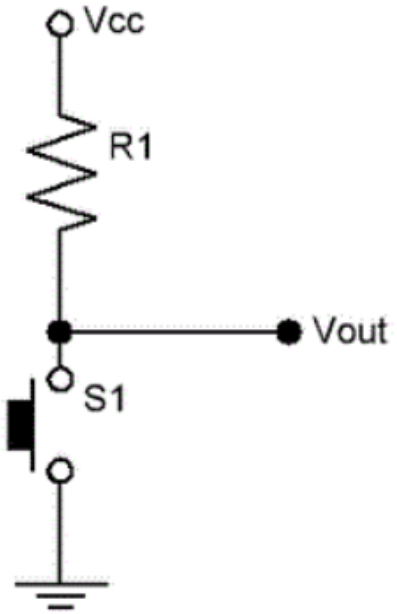


```
#include "stm32f7xx.h"
int main(void){

  RCC -> AHB1ENR = 0X6; //HABILITAR LOS PUERTOS B Y C

  GPIOB -> MODER = 0X10004001; //
  GPIOB -> OTYPER = 0X0; //PUSH PULL
  GPIOB -> OSPEEDR = 0x55555555; //VELOCIDAD MEDIA
  GPIOB -> PUPDR = 0X10004001; // NO PULL UP
  GPIOC -> MODER = 0; //


  while(1){
      if((GPIOC -> IDR&0X2000)==0X2000){
      for(int i=0;i< 100000; i++);
      GPIOB -> ODR ^=0x4081;
          }
    }
  }
```



Pull Down

```c
#include "stm32f7xx.h"
int main(void){

  RCC -> AHB1ENR = 0X6; //HABILITAR LOS PUERTOS B Y C

  GPIOB -> MODER = 0X10004001; //
  GPIOB -> OTYPER = 0X0; //PUSH PULL
  GPIOB -> OSPEEDR = 0x55555555; //VELOCIDAD MEDIA
  GPIOB -> PUPDR = 0X10004001; // NO PULL UP
  GPIOC -> MODER = 0; //

  //RESISTENCIA EN PULL-DOWN
  while(1){
      if((GPIOC -> IDR&0X2000)==0){
      for(int i=0;i< 100000; i++);
      GPIOB -> ODR ^=0x4081;
          }
    }
  }
```

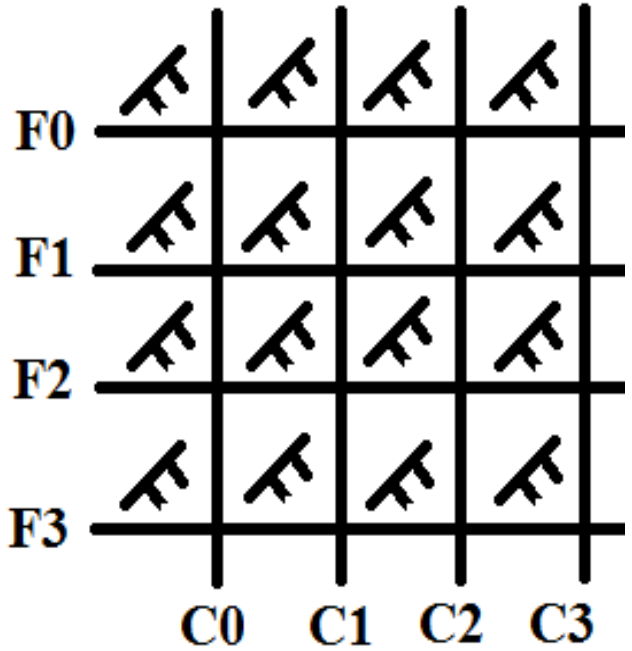


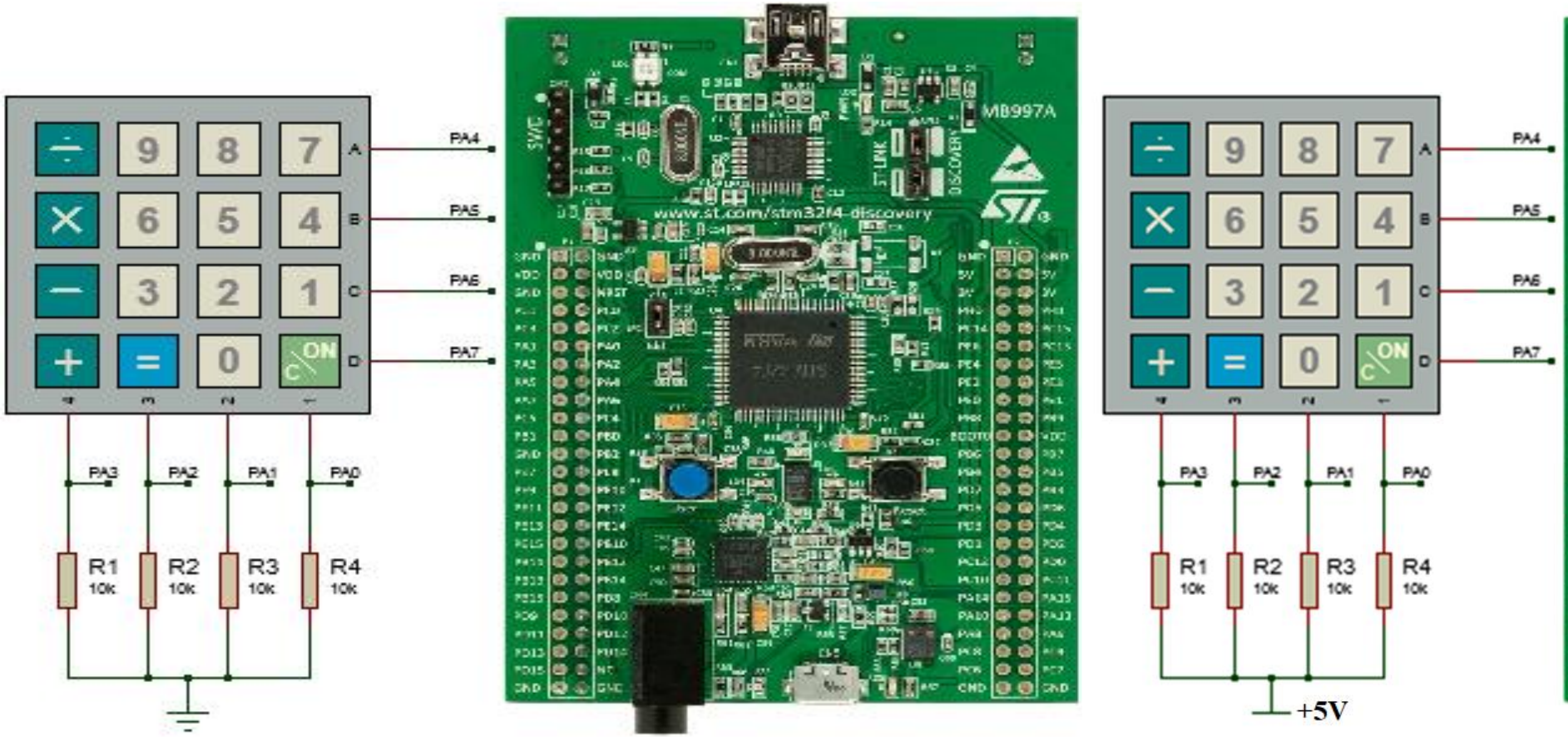
Pull Up

```c
#include "stm32f7xx.h"
int main(void){
  int a=0;
  RCC -> AHB1ENR = 0X6; //HABILITAR LOS PUERTOS B Y C

  GPIOB -> MODER = 0X10004001; //
  GPIOB -> OTYPER = 0X0; //PUSH PULL
  GPIOB -> OSPEEDR = 0x55555555; //VELOCIDAD MEDIA
  GPIOB -> PUPDR = 0X10004001; // NO PULL UP
  GPIOC -> MODER = 0; //

  //RESISTENCIA EN PULL-DOWN
  while(1){
      while((GPIOC -> IDR&0X2000)==0X2000){
      for(int i=0;i< 100000; i++);
      a++;
          }
    }
  }
```

```c
#include "stm32f7xx.h"
int main(void){
  int a=0;
  RCC -> AHB1ENR = 0X6; //HABILITAR LOS PUERTOS B Y C

  GPIOB -> MODER = 0X10004001; //
  GPIOB -> OTYPER = 0X0; //PUSH PULL
  GPIOB -> OSPEEDR = 0x55555555; //VELOCIDAD MEDIA
  GPIOB -> PUPDR = 0X10004001; // NO PULL UP
  GPIOC -> MODER = 0; //

  //RESISTENCIA EN PULL-DOWN
  while(1){
      if((GPIOC -> IDR&0X2000)==0X2000){
      for(int i=0;i< 100000; i++);
        while((GPIOC -> IDR&0X2000)==0X2000);
      a++;
          }
    }
  }
```

TECLADO MATRICIAL



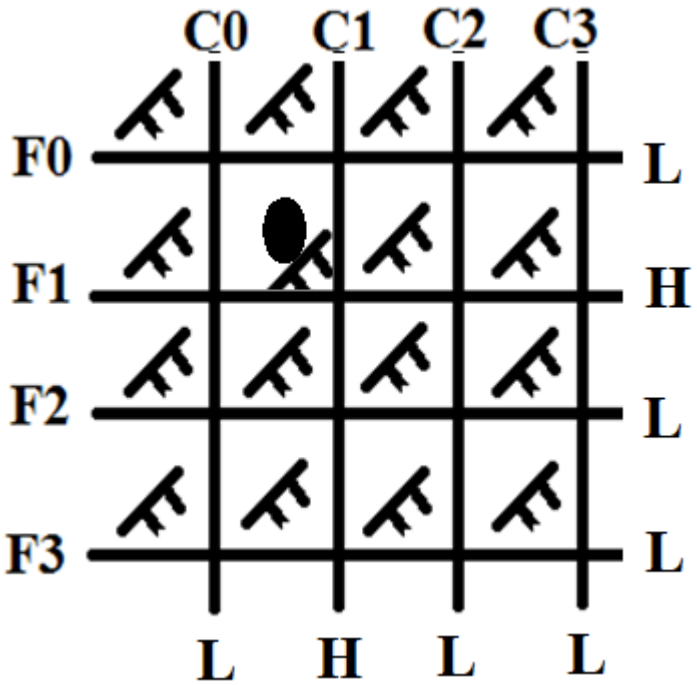

9
8
7
A
6
5
4
B
3
2
1
C
0
ON
C
D
PA4
PA5
PA6
PA7
PA3
PA2
PA1
PA0
R1
10k
R2
10k
R3
10k
R4
10k
MB997A
www.st.com/stm32f4-discovery
+5V

Barrido matricial.

Evaluar celda por celda para determinar si se ha activado alguna.

Ejemplo: F1C1 LHLLLHLL o puede ser

LHLLLLHL



f0f1f2f3c0c1c2c3	codigo	signif
10000000	128	act f0
01000000	64	act f1
00100000	32	act f2
00010000	16	act f3
10000001	129	# 1
10000010	130	# 2
10000100	132	# 3
10001000	136	# 4
01000001	65	# 5
01000010	66	# 6
01000100	68	# 7
01001000	72	# 8
00100001	33	# 9
00100010	34	# 10
00100100	36	# 11
00101000	40	# 12
00010001	17	# 13
00010010	18	# 14
00010100	20	# 15
00011000	24	# 0

f0c0f1c1f2c2f3c3	codigo	signif
10000000	128	act f0
00100000	32	act f1
00001000	8	act f2
00000010	2	act f3
10000001	?	# 1
10000010	?	# 2
10000100	?	# 3
10001000	?	# 4
01000001	?	# 5
01000010	?	# 6
01000100	?	# 7
01001000	?	# 8
00100001	?	# 9
00100010	?	# 10
00100100	?	# 11
00101000	?	# 12
00010001	?	# 13
00010010	?	# 14
00010100	?	# 15
00011000	?	# 0

```c
#include <stdio.h>
#include "stm32f7xx.h"
int numeros[10]={126,48,109,121,51,91,95,112,127,123};
int a,cl;
int teclado(void){
GPIOB -> ODR ^=128;
for(int i = 0; i < 100000; i++){};
cl=(GPIOB -> IDR)&0X000000FF;
if(cl==129){return 1;}
else if(cl==130){return 2;}
else if(cl==132){return 3;}
else if(cl==136){return 4;}
GPIOB -> ODR ^=64;
for(int i = 0; i < 100000; i++){};
cl=(GPIOB -> IDR)&0X000000FF;
if(cl==65){return 5;}
else if(cl==66){return 6;}
else if(cl==68){return 7;}
else if(cl==72){return 8;}
GPIOB -> ODR ^=32;
for(int i = 0; i < 100000; i++){};
cl=(GPIOB -> IDR)&0X000000FF;
if(cl==33){return 9;}
else if(cl==34){return 10;}
else if(cl==36){return 11;}
else if(cl==40){return 12;}
GPIOB -> ODR ^=16;
for(int i = 0; i < 100000; i++){};
cl=(GPIOB -> IDR)&0X000000FF;
if(cl==13){return 1;}
else if(cl==18){return 14;}
else if(cl==20){return 15;}
else if(cl==24){return 0;}
}
```

```
int main(void){

  RCC -> AHB1ENR = 0X6; //HABILITAR LOS PUERTOS B Y C

  GPIOB -> MODER = 0X00005500; //
  GPIOB -> OTYPER = 0X0; //PUSH PULL
  GPIOB -> OSPEEDR = 00005555; //VELOCIDAD MEDIA
  GPIOB -> PUPDR = 0; // NO PULL UP

    GPIOC -> MODER = 0X00005555; //
  GPIOC -> OTYPER = 0X0; //PUSH PULL
  GPIOC -> OSPEEDR = 00005555; //VELOCIDAD MEDIA
  GPIOC -> PUPDR = 0; // NO PULL UP

  while(1){

      for( a = 0; a < 10; a++){};

        a=teclado();
        GPIOC-> ODR ^= numeros[a];

    }
  }
```

TAREA

Desarrollar un programa que sume o multiplique dos números ingresados por teclado matricial y muestre el resultados por un puerto en que hay 8 leds de visualización.