/\*

\* main.c

\*

\* Created: 9/1/2023 12:21:16 PM

\* Author: 2161908

\*/

#define *F\_CPU* 1000000UL

#include <util/delay.h>

#include <avr/io.h>

#include <avr/sfr\_defs.h>

#define DDR\_SPI DDRB

#define CS 2

#define CS\_SoftCode 1

#define MOSI 3

#define MISO 4

#define SCK 5

void SPI\_SlaveInit(void)

{

/\* Set MISO output, all others input \*/

DDR\_SPI = (1<<MISO);

/\* Enable SPI \*/

SPCR = (1<<SPE);

}

char SPI\_SlaveReceive(void)

{

/\* Wait for reception complete \*/

while(!(SPSR & (1<<SPIF)))

;

/\* Return Data Register \*/

return SPDR;

}

void Blink(){

PORTC |= (1<<0);

*\_delay\_ms*(1000);

PORTC &= ~(1<<0);

*\_delay\_ms*(1000);

PORTC |= (1<<1);

*\_delay\_ms*(1000);

PORTC &= ~(1<<1);

*\_delay\_ms*(1000);

}

int main(void)

{

/\* initialiser le slave SPI \*/

SPI\_SlaveInit();

/\* mettre les deux portes en sortie \*/

DDRC |= (1<<0)|(1<<1);

/\* test de lumière pour confirmer \*/

Blink();

/\* pour envoyer plusieurs char test \*/

char transmit[] = {0x21,0x22,0x33};

while (1)

{

char receive = SPI\_SlaveReceive();

if(receive == 0x55){

SPDR = 0x51;

PORTC |= (1<<0);

}

if(receive == 0x33){

SPDR = 0x31;

PORTC |= (1<<1);

}

if(receive == 0x44){

SPDR = 0xAA;

PORTC &= ~(1<<0);

PORTC &= ~(1<<1);

break;

}

}

return(0);

}