# Programas C

```
#define TRUE 1
#define FALSE 0
#define BufSize 127
#define C 0
#define B 1

int getnum(char* x);

void setup()
{
  // start serial port at 9600 bps:
  Serial.begin(9600);
  DDRC=B00000000;
  PORTC=B11111111;
  DDRB=B11111111;
  PORTB=B00000000;
}

void loop()
{
  int i;
  int Entry[2];
  int Past[2];
  int Hist[2];
  char IncomingByte;
  char State[BufSize];


  Serial.flush();
  //INIC

  for( Past[C] = Entry[C], Past[B] = Entry[B]; TRUE; Hist[C] = Entry[C], Hist[B] = Entry[B]){
    //ENTRADA Portas
    Entry[C]=PINC;
    //ENTRADA Serial
    if(Serial.available() > 0){
      delay(25);//wait for incoming data.
      for(i = 0; IncomingByte = Serial.read(); i++){
            if((IncomingByte == '\r') || (IncomingByte == '\n')){
              State[i] = '\0';
              Serial.flush();
              break;
            }else{
              State[i]=IncomingByte;
            }
      }
      Entry[B] = getnum(&State[0]);
    }

    if((Entry[C] == Hist[C]) && (Entry[B] == Hist[B]))
      continue;
    for( ; TRUE; Past[C] = Entry[C], Past[B] = Entry[B]){
      if((Entry[C] == Past[C]) && (Entry[B] == Past[B]))
            break;

      /**Processing***/
      if(Entry[C] != Past[C]){
            if(!(Serial.available() > 0)){
              //leituras do microcontrolador.
              Serial.println(Entry[C],DEC);
              delay(10);
            }
      }

      if(Entry[B] != Past[B]){
              PORTB = Entry[B];
      }
    }
  }
}
//have to press reset in learning mode always.
/***FUNCTIONS***/
int getnum(char* x)
{
  int num;
  if(sscanf(x,"%d",&num) != 0){
    if (num == NULL)
      num = 0;
    return num;
  }else{
    return 0;
  }
}
//char *X, X is a variable that stores a fisical
//address with a cast type char.
//getnum works with string terminating with only NULL or '\0'
//my style
//Lesson always be flexible and except and obey the good practices,
// without any desobidience, and for extra culture if desired try to
// clarify the why things are as they are. Never be stuborn.
```

```
#define TRUE 1
#define FALSE 0
#define column 4
#define lines 36

void setup()
{
  // start serial port at 9600 bps:
  Serial.begin(9600);
  DDRC=B00000000;
  PORTC=B11111111;
  DDRB=B11111111;
  PORTB=B00000000;
}

void loop()
{
  int i, l, c;
  int data[5];
  int Hist[5];
  int count;

  //inic key
  data[0]=0;
  data[1]=63;
  data[2]=0;
  PORTB = data[2];
  //mem prepared for depth 2 in FSM (finie state machine).
  int mem[5][36]=
  {
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0      },//pres
    { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0      },//tran pas
    { 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0      },//past
    {63,53,52,49,53,63,49,52,61,55,48,60,51,54,57,60,62,59,51,50,56,48      }, //tran pre
    { 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1      } //present
  };

  /*******************CICLOS DE MAQUINA*********************/
  for( Hist[0] = data[0], Hist[1] = data[1], Hist[2] = data[2]; TRUE; Hist[3] = data[3]){

      data[3] = PINC;

    delay(60);


    if(!(Serial.available()>0)){
      Serial.print(data[3]);
      Serial.println(count);
    }

    //timer setup
    if(data[4]==1){
      count++;
    }else{
      count=0;
    }
    //catch timer
    if(count==15000){
      data[4]=0;
      PORTB=data[4];
            //update
      data[0]=data[2];
      data[1]=data[3];
      data[2]=data[4];
      Hist[0]=data[0];
      Hist[1]=data[1];
      Hist[2]=data[2];
      Hist[4]=data[4];
    }
    //end timer

    /*****/
    if(data[3] == Hist[3])
      continue;
//use this for more depth state machine or c < 2.
//    if(data[1] == data[3])
//      continue;
    /*****/
    /**********************Search and apply changes*********************/
    for( l = 0, c = 0; l < lines   ; l++){
      if(c >= column){
        l--;
        data[4] = mem[c][l];
        PORTB = data[4];

        //update
        data[0]=data[2];
        data[1]=data[3];
        data[2]=data[4];
        Hist[0]=data[0];
        Hist[1]=data[1];
        Hist[2]=data[2];
        Hist[4]=data[4];

        break;
      }
      //startup c=2, c=0, for more precise.
      for(c=2; c < column; c++){
        if(data[c] == mem[c][l]){
          continue;
        }else{
          break;
        }
      }
    }
    /*******************************************************************************/




    /*******************************************************************************/
  }
}
//Nao existe futuro apenas present proximo.
// there is aproblem with comunication of the arduino inputs due to input beeing 8bit word.
//The more in depth the state machine is the more obidiente and dumb it becomes, because
//it will have to specify all possible cases.
```

```
#define TRUE 1
#define FALSE 0
#define BufSize 127
#define A 0
#define C 1

int getnum(char* x);

void setup()
{
  // start serial port at 9600 bps:
  Serial.begin(9600);
  DDRA=B00000000;
  PORTA=B11111111;
  DDRC=B11111111;
  PORTC=B00000000;
}

void loop()
{
  int i;
  int Entry[2];
  int Past[2];
  int Hist[2];
  char IncomingByte;
  char State[BufSize];


  Serial.flush();
  //INIC

  for( Past[A] = Entry[A], Past[C] = Entry[C]; TRUE; Hist[A] = Entry[A], Hist[C] = Entry[C]){
    //ENTRADA Portas
    Entry[A]=PINA;
    //ENTRADA Serial
    if(Serial.available() > 0){
      delay(25);//wait for incoming data.
      for(i = 0; IncomingByte = Serial.read(); i++){
            if((IncomingByte == '\r') || (IncomingByte == '\n')){
              State[i] = '\0';
              Serial.flush();
              break;
            }else{
              State[i]=IncomingByte;
            }
      }
      Entry[C] = getnum(&State[0]);
    }

    if((Entry[A] == Hist[A]) && (Entry[C] == Hist[C]))
      continue;
    for( ; TRUE; Past[A] = Entry[A], Past[C] = Entry[C]){
      if((Entry[A] == Past[A]) && (Entry[C] == Past[C]))
            break;

      /**Processing***/
      if(Entry[A] != Past[A]){
            if(!(Serial.available() > 0)){
              //leituras do microcontrolador.
              Serial.println(Entry[A],DEC);
              delay(10);
            }
      }

      if(Entry[C] != Past[C]){
              PORTC = Entry[C];
      }
    }
  }
}
//have to press reset in learning mode always.
/***FUNCTIONS***/
int getnum(char* x)
{
  int num;
  if(sscanf(x,"%d",&num) != 0){
    if (num == NULL)
      num = 0;
    return num;
  }else{
    return 0;
  }
}
//char* X, X is a variable that stores a fisical
//address with a cast type char.
//getnum works with string terminating with only NULL or '\0'
//my style
```

```
#define TRUE 1
#define FALSE 0
#define column 4
#define lines 36

void setup()
{
 // start serial port at 9600 bps:
 Serial.begin(9600);
 DDRA=B00000000;
 PORTA=B11111111;
 DDRC=B11111111;
 PORTC=B00000000;
}

void loop()
{
 int i, l, c;
 int data[5];
 int Hist[5];
 int count;

 //inic key
 data[0]=0;
 data[1]=255;
 data[2]=0;
 PORTB = data[2];
 //mem prepared for depth 2 in FSM (finie state machine).
 int mem[36][5]=
 {
  { 0, 0, 0,255, 0},
  { 0, 0, 0,254, 1},
  { 0, 0, 1,253, 0},
  { 0, 0, 1,254, 2},
  { 0, 0, 2,254, 1}
 };

 /*******************CICLOS DE MAQUINA*********************/
 for( Hist[0] = data[0], Hist[1] = data[1], Hist[2] = data[2]; TRUE; Hist[3] = data[3]){

    data[3] = PINA;

   delay(60);


   if(!(Serial.available()>0)){
    Serial.print(data[3]);
```

```
      Serial.print(", ");
      Serial.println(count);
    }

    //timer setup
    if(data[4]==1){
      count++;
    }else{
      count=0;
    }
    //catch timer
    if(count==100){
      data[4]=0;
      PORTC=data[4];
      //update
      data[0]=data[2];
      data[1]=data[3];
      data[2]=data[4];
      Hist[0]=data[0];
      Hist[1]=data[1];
      Hist[2]=data[2];
      Hist[4]=data[4];
    }
    //end timer

    /*****/
    if(data[3] == Hist[3])
      continue;
//use this for more depth state machine or c < 2.
//    if(data[1] == data[3])
//      continue;
    /*****/
    /*********************Search and apply changes*********************/
    for( l = 0, c = 0; l < lines   ; l++){
      if(c >= column){
        l--;
        data[4] = mem[l][c];
        PORTC = data[4];

        //update
        data[0]=data[2];
        data[1]=data[3];
        data[2]=data[4];
        Hist[0]=data[0];
        Hist[1]=data[1];
        Hist[2]=data[2];
        Hist[4]=data[4];

        break;
      }
      //startup c=2, c=0, for more precise.
      for(c=2; c < column; c++){
        if(data[c] == mem[l][c]){
          continue;
        }else{
          break;
        }
      }
    }
    /*****FALL THREW*******************************************************/





    /**********************************************************************/
  }
}
//Nao existe futuro apenas present proximo.
// there is aproblem with comunication of the arduino inputs due to input beeing 8bit word.
//The more in depth the state machine is the more obidiente and dumb it becomes, because
//it will have to specify all possible cases.
//char *X, X is a variable that stores a fisical
//address with a cast type char.
//getnum works with string terminating with only NULL or '\0'
//my style
//Lesson always be flexible and except and obey the good practices,
// without any desobidience, and for extra culture if desired try to
// clarify the why things are as they are. Never be stuborn.
```

```
/***sergio manuel salazar dos santos 934805603***/
/***Rua do relogio 268, 4770-245 Joane, Vila Nova de famalicao, Braga, Portugal***/
/***CABECALHO Libraries***/
//this program is a learning finite state machine.
#include "creation_7.h"

/***MAIN***/
int main(int argc, char* argv[])
{

//capture prog arguments.
if(argc < 2){
	printf("Enter the board use duemilanove or mega ? \n");
	return 0;
}else if(argc < 3){
	printf("Enter a file name please \n");
	return 0;
}else if(argc < 4){
	printf("Enter learning mode on or off \n");
	return 0;
}

printf("FILE -> %s\n", __FILE__);
printf("DATE -> %s TIME -> %s\n", __DATE__, __TIME__);
printf("DATE -> %d\n", __LINE__);
printf("double -> %d\n", Double(5));
//printf("Pow -> %f\n", Pow(2,5));
printf("argv[0] (Progname) -> %s	size: %d		\n", argv[0], strlen(argv[0]));
printf("argv[1] (Board)-> %s size: %d		\n", argv[1], strlen(argv[1]));
printf("argv[2] (Filename)-> %s		size: %d		\n", argv[2], strlen(argv[2]));
printf("argv[3] (Learnmode) -> %s	size: %d		\n", argv[3], strlen(argv[3]));

//make decisions relative income arguments.
if(strcmp(argv[1], "mega")==0){
	DEVICE = Putstr(DEVICE_1);
}else if(strcmp(argv[1], "duemilanove")==0){
	DEVICE = Putstr(DEVICE_2);
}else{
	printf("Board options mega or duemilanove \n");
	return 0;
}

if(strcmp(argv[3], "on")==0){
	LEARN=1;
}else if(strcmp(argv[3], "off")==0){
	LEARN=0;
}else{
        printf("LEARN mode options on or off\n");
        return 0;
}


/***Internal Variables***/
int errno;

/***portcomunication setup file descriptor***/
int fd;
int c, ires, ores;

fd = open(DEVICE, O_RDWR | O_NOCTTY | O_NDELAY);
printf("file descriptor: %d\n",fd);
if(fd < 0)
        goto EXIT_2;

//com prepare
struct termios oldtio,newtio;
tcgetattr(fd, &oldtio);
bzero(&newtio, sizeof(newtio));

/*
BAUDRATE: Set bps rate. You could also use cfsetispeed and cfsetospeed.
CRTSCTS : output hardware flow control (only used if the cable has
      all necessary lines. See sect. 7 of Serial-HOWTO)
CS8     : 8n1 (8bit,no parity,1 stopbit)
CLOCAL  : local connection, no modem control
CREAD   : enable receiving characters
*/
//newtio.c_cflag = BAUDRATE | CRTSCTS | CS8 | CLOCAL | CREAD;
newtio.c_cflag = BAUDRATE | CS8 | CLOCAL | CREAD;

/*
IGNPAR  : ignore bytes with parity errors
ICRNL   : map CR to NL (otherwise a CR input on the other computer
      will not terminate input)
otherwise make device raw (no other input processing)
*/
//newtio.c_iflag |= (IGNPAR | ICRNL);
newtio.c_iflag |= (INPCK | ISTRIP);
//newtio.c_iflag |= INPCK;

/*
Raw output. first option.
*/
//newtio.c_oflag |= 0;
//newtio.c_oflag |= (OPOST | ONLCR);
newtio.c_oflag |= OPOST;
```

```
//newtio.c_oflag &= ~OPOST;

/*
ICANON  : enable canonical input
disable all echo functionality, and don't send signals to calling program
*/
newtio.c_lflag = ICANON;

/*
initialize all control characters
default values can be found in /usr/include/termios.h, and are given
in the comments, but we don't need them here
*/
newtio.c_cc[VINTR]    = 0;     /* Ctrl-c */
newtio.c_cc[VQUIT]    = 0;     /* Ctrl-\ */
newtio.c_cc[VERASE]   = 0;     /* del */
newtio.c_cc[VKILL]    = 0;     /* @ */
newtio.c_cc[VEOF]     = 4;     /* Ctrl-d */
newtio.c_cc[VTIME]    = 0;     /* inter-character timer unused */
newtio.c_cc[VMIN]     = 1;     /* blocking read until 1 character arrives */
newtio.c_cc[VSWTC]    = 0;     /* '\0' */
newtio.c_cc[VSTART]   = 0;     /* Ctrl-q */
newtio.c_cc[VSTOP]    = 0;     /* Ctrl-s */
newtio.c_cc[VSUSP]    = 0;     /* Ctrl-z */
newtio.c_cc[VEOL]     = 0;     /* '\0' */
newtio.c_cc[VREPRINT] = 0;     /* Ctrl-r */
newtio.c_cc[VDISCARD] = 0;     /* Ctrl-u */
newtio.c_cc[VWERASE]  = 0;     /* Ctrl-w */
newtio.c_cc[VLNEXT]   = 0;     /* Ctrl-v */
newtio.c_cc[VEOL2]    = 0;     /* '\0' */

/*
now clean the modem line and activate the settings for the port
*/
tcflush(fd, TCIFLUSH);
if(tcsetattr(fd, TCSANOW, &newtio) != 0)
        goto EXIT_3;

/*******************************************/
/***nanosleep***/
//alivea procesador.
struct timespec* timer_1;
timer_1 = calloc(1, sizeof(struct timespec));
timer_1->tv_sec = 0;
timer_1->tv_nsec = 10000000;
struct timespec* timer_2;
timer_2 = calloc(1, sizeof(struct timespec));
timer_2->tv_sec = 1;
timer_2->tv_nsec = 0;
```

```
/******/
/***Variables***/
int i, j, n;
buildingx leitura;
buildingx memory;
pastx hist;
char ordem[SEND];

/***File that stores de structs***/
FILE* fsm;
fsm=fopen(argv[2], "a+");
if(fsm == NULL)
        goto EXIT_1;

//Inicialize the primary struct.
printf("struct size:%d\n\7", sizeof(leitura));
//saida ou estado inicial.

strcpy(&leitura.key[0][0], "\r\n");
strcpy(&leitura.key[1][0], "\r\n");
strcpy(&leitura.who[0][0], "\r\n");
strcpy(&leitura.who[1][0], "\r\n");

//delay.
nanosleep(timer_1, NULL);

printf("        Press reset button on target!!\n");
perror("status ");

//The main program starts here.
/***CICLOS MAQUINA***/
for(printf("WELCOME!!!\n"); TRUE; strcpy( &hist.registry[1][0], &leitura.key[1][0])){

        //ENTRADAS.
        //Entrada Serial
        ires=read(fd, &leitura.key[1][0], buf_size);

        //printf("%s", &leitura.key[1][0]);

        if(ires > 0){
                for(i=0; i < buf_size; i++){
                        if(leitura.key[1][i] == '\n'){
                                i++;
                                break;
                        }
                }
                leitura.key[1][i] = '\0';
        }else{
                //perror("status ");
```

```
			nanosleep(timer_1,NULL);
			continue;
		}

		if(strcmp( &leitura.key[1][0], &hist.registry[1][0])==0)
			continue;
		printf("\nleitura : %d ->  %s at %s", ires, &leitura.key[1][0],__TIME__);
		if(strcmp(&leitura.key[0][0],&leitura.key[1][0])==0)
			continue;
		/***************************************************************************
******************/

			//Search Engine
			for(i=0, rewind(fsm); fread(&memory, sizeof(buildingx), 1, fsm); i++){
				//found we can choose the depth of FSM here.
				if(
					strcmp(&leitura.key[0][0], &memory.key[0][0])==0 &&
					strcmp(&leitura.key[1][0], &memory.key[1][0])==0 &&
					strcmp(&leitura.who[0][0], &memory.who[0][0])==0	){

					printf("struct = %d ", i);
					printf("out -> %s", &memory.who[1][0]);
					//preset
					strcpy(&leitura.key[0][0],&leitura.key[1][0]);
					strcpy(&leitura.who[0][0], &memory.who[1][0]);

					if(strcmp(&leitura.who[0][0], "quit\r\n")==0){
						ores = write(fd, "0\r\n", 4*sizeof(char));
						goto EXIT_1;
					}else{
						//Get number string

						if(sscanf(&leitura.who[0][0], "%[0-9]", ordem) == 0)
							strcpy(ordem,"0");
						strcat(ordem,"\r\n");

						//printf("ordem -> %s", ordem);

						ores = write(fd,ordem, strlen(ordem));
						break;
					}
				}
				//error
				if(ferror(fsm)){
					perror("status ");
					errno = 0;
					break;
				}
			}
			//procedures executed only if in LEARN mode on.
			if(feof(fsm)){
				if(LEARN){
					printf("--> new data\n");
					infor(&leitura.who[1][0], buf_size, stdin);

					if(strcmp(&leitura.who[1][0], "exit\r\n") == 0){
						goto EXIT_1;
					}else{
						//Get number string

						if(sscanf(&leitura.who[1][0], "%[0-9]", ordem) == 0)
							strcpy(ordem,"0");
						strcat(ordem,"\r\n");

						ores = write(fd, ordem, strlen(ordem));
					}

					fseek(fsm, 0, SEEK_END);
					fwrite(&leitura, sizeof(buildingx), 1, fsm);
					//preset
					strcpy(&leitura.key[0][0], &leitura.key[1][0]);
					strcpy(&leitura.who[0][0], &leitura.who[1][0]);

					printf("done!\n");
					continue;
				}else{
					printf("Unknown input!\n");
				}
		}
/
*******************************************************************************
************/
}
//EXITS in various points of the program.
EXIT_1:
	nanosleep(timer_2,NULL);
	/* restore the old port settings */
	tcsetattr(fd, TCSANOW, &oldtio);
	close(fd);
	free(DEVICE);
	free(timer_1);
	free(timer_2);
	fclose(fsm);
	return 0;
EXIT_2:
	perror(DEVICE);
	free(DEVICE);
```

```
        exit(-1);
        return 0;
EXIT_3:
        /* restore the old port settings */
        tcsetattr(fd, TCSANOW, &oldtio);
        close(fd);
        free(DEVICE);
        return 0;

}//main

//new aproach and I prefer it.
//a thing is returning a pointer.
//another is writting to an address.
//beautifull work.
//pointer are easy but take work in allocating
//memory all the time. function ReadConsole and Putstr demonstrates just that.
//obra de arte.
//magic formula, address=pow(2,0)*input+pow(2,8)*output;
//Going to try to save the file inside the microcontroller so that it will
//run independently without Serial comunication.
//There is no such thing as future only coming present.
//search for more bugs.
//just for the fun of it going to try a new aproach and consider this stable and finished.
//New capabilities can be added later.
```

```
/*creation_7.h*/
#ifdef _CREATION_7_H_
#else
	#define _CREATION_7_H_
//PROTOTYPE
/**sergio manuel salazar dos santos 934805603**/
/***Libraries***/
// fopen perror fread fwrite feof fseek ferror fclose rewind scanf sscanf getchar scanf fscanf
// strncpy sscanf
#include <stdio.h>
// calloc free realloc
#include <stdlib.h>
// strcpy strcmp strcat memcmp
#include <string.h>
// termios tcflush
#include <termios.h>
// nanosleep sleep
#include <time.h>
// tcflsuh read write close
#include <unistd.h>
// perror
#include <errno.h>
// open
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
//assert
#include <assert.h>
//offsetof
#include <stddef.h>
//__fpurge
//#include <stdio_ext.h>
//#include <netdb.h>
//#include <netinet/in.h>
//#include <stdbool.h>
//#include <ctype.h>
//#include <limits.h>
//#include <semaphore.h>
//#include <pthread.h>
//#include <math.h>
//#include <signal.h>
//#include <sys/mman.h>
//#include <sys/wait.h>
//#include <sys/time.h>
//#include <sys/resource.h>
#include <sys/dir.h>
//#include <sys/socket.h>
```

```
//#include <sys/un.h>


/***MACROS***/
#define TRUE 1
#define FALSE 0
#define BAUDRATE B9600
//arduino MEGA
#define DEVICE_1 "/dev/serial/by-id/usb-FTDI_FT232R_USB_UART_A600aiS5-if00-port0"
//arduino DUEMILANOVE
#define DEVICE_2 "/dev/serial/by-id/usb-FTDI_FT232R_USB_UART_A9007Qwg-if00-port0"
#define _POSIX_SOURCE 1
#define buf_size 32
#define SEND 16
#define Double(x) (2*(x))

/***Gloabal Variables***/
int LEARN;
char* DEVICE;

/***PROTO***/
typedef struct {
	char registry[2][buf_size];
}pastx;

typedef struct {
	char key[2][buf_size];
	char who[2][buf_size];
}buildingx;

#endif

/***FUNCTION TITLES***/
#include "creation_7_func.h"
```

```
//PROTOTYPE
/**sergio manuel salazar dos santos 934805603**/

#include "creation_7.h"

/*FUNCOES*/
/*****Putstr*******/
char* Putstr(char* str)
{
	int i; char* ptr;
	ptr = (char*)calloc(strlen(str), sizeof(char));
	if(ptr == NULL){
		perror("NULL!\n");
		return NULL;
	}
	for(i=0; (ptr[i] = str[i]); i++)
	{
		if(ptr[i] == '\0')
			break;
	}
	return (ptr);
}

/***ReadConsole***/
char* ReadConsole(FILE* stream)
{
	int i, NBytes;
	char caracter;
	char* value=NULL;
	for(i=0, NBytes=8; (caracter=getc(stream)) != EOF; i++){
		if((i==NBytes) | (i==0)){
			NBytes=2*NBytes;
			value=(char*)realloc(value, NBytes*sizeof(char));
			if(value==NULL)
				perror(value);
		}
		*(value+i)=caracter;
		if(caracter=='\n'){
			*(value+i)='\0';
			break;
		}
	}
	return value;
}

/***getnum***/
int getnum(int min, int max)
```

```
{
	int num;
	for(num=0; (scanf("%d",&num)==0) || num<min || num>max ; getchar()){
		perror("loop status");
	}
	return num;
}

/*********fillvec************/
int* fillvec(int num, int size)
{
	int* x;
	int i;
	if(size > 0){
		x=calloc(size, sizeof(int));
		for(i=0; i<=size; i++)
		{
			x[i]=num;
			//printf("x[%d]-> %d\n",i,x[i]);//troubleshooting
		}
	}else{
		return NULL;
	}
	return x;
}

/*******ReadFiletoMem*******/
void* ReadFiletoMem(void* datatype, FILE* filename)
{
	int i , n;
	fseek(filename, 0, SEEK_SET);
	datatype=calloc(1,sizeof(*datatype));
	for(i=0, n=1; fread((datatype+i), sizeof(*datatype), 1, filename); datatype=realloc(datatype,
n*sizeof(*datatype))){
		i++;
		n++;
		if(feof(filename))
			break;
		if(ferror(filename)){
			perror("status:");
			return NULL;
		}
	}
	return datatype;
}

/**GetChar()**/
unsigned char GetChar()
{
	unsigned char x;
	unsigned char value;
	for(value=getchar(); x!='\n'; x=getchar());
	if(value=='\0')
		return 1;
	x='0';
	return value;
}

/******/
//sintaxe muito muito importante, mais importante doque a semantica.
//I learn allot reading other peoples code.
char* ReadConsoleSer(FILE* stream)
{
	int i, NBytes;
	char caracter;
	char* value=NULL;
	for(i=0, NBytes=8; (caracter=getc(stream)) != EOF;i++){
		if((i==NBytes) | (i==0)){
			NBytes=2*NBytes;
			value=(char*)realloc(value,NBytes*sizeof(char)+2);
			if(value==NULL)
				perror(value);
		}
		*(value+i)=caracter;
		if(caracter=='\n'){
			*(value+i)='\r';
			i++;
			*(value+i)='\n';
			i++;
			*(value+i)='\0';
			break;
		}
	}
	return value;
}

/******/
int getnumber(char* x)
{
  int num;
  if(sscanf(x, "%d", &num)!=0)
    return num;
  else
    return 0;
}

/***infor***/
int infor(char* inf, int Size_Inf, FILE* stream){
	int n;
	char* a;
	a=calloc(1024, sizeof(char));
	while((n=fscanf(stream, "%s", a)) != 0){
		if(strlen(a) > (Size_Inf-3)){
			printf("overflow retry.\n");
			continue;
		}
		strncpy(inf, a, (Size_Inf-3));
		strcat(inf, "\r\n");
		break;
	}
	free(a);
	return n;
}

/******/
```

```
/*creation_7_func.h*/
#ifdef _CREATION_7_FUNC_H_
#else
	#define _CREATION_7_FUNC_H
//PROTOTYPE
/***FUNCTION TITLES***/
/***Coloca uma string numa variavel tipo apontador allocando tamanho automatico***/
char* Putstr(char* str);
/***Lê stdin e aloca automaticamente espaço em memória***/
char* ReadConsole();
/***lê apenas numeros de uma string***/
int getnum(int min, int max);
/***preenche vector de tamanho size por num***/
int* fillvec(int num, int size);
/***ReadFiletoMem***/
void* ReadFiletoMem(void* datatype,FILE* filename);
/***GetChar***/
unsigned char GetChar();
/***ReadConsoleR***/
char* ReadConsoleSer(FILE* stream);
/***gets the number out of a string***/
int getnumber(char* x);
/*******/
int infor(char* inf, int Size_Inf, FILE* stream);
/*******/
#endif
```

```
CC=gcc
LIB=-L./

all:resultado

resultado:creation_7.o creation_7_func.o
	${CC} creation_7.o creation_7_func.o -Wall -lm -o FSM.exe ${LIB}

resultado.o:creation_7.c
	${CC} -c creation_7.c -Wall -lm -o creation_7.o ${LIB}

resultado_func.o:creation_7_func.c
	${CC} -c creation_7_func.c -Wall -lm -o creation_7_func.o ${LIB}

clean.o:
	rm *.o *.exe
```

```
#include<avr/io.h>
#define TRUE 1
#define FALSE 0
#define BufSize 127
#define C 0
#define B 1

int getnum(char* x);

void setup()
{
  // start serial port at 9600 bps:
  Serial.begin(9600);
  DDRC=B00000000;
  PORTC=B11111111;
  DDRB=B11111111;
  PORTB=B00000000;
}

void loop()
{
  int i;
  int Entry[2];
  int Past[2];
  int Hist[2];
  char IncomingByte;
  char State[BufSize];


  Serial.flush();
  //INIC

  for( Past[C] = Entry[C], Past[B] = Entry[B]; TRUE; Hist[C] = Entry[C], Hist[B] = Entry[B]){
    //ENTRADA Portas
    Entry[C]=PINC;
    //ENTRADA Serial
    if(Serial.available() > 0){
      delay(25);//wait for incoming data.
      for(i = 0; IncomingByte = Serial.read(); i++){
            if((IncomingByte == '\r') || (IncomingByte == '\n')){
              State[i] = '\0';
              Serial.flush();
              break;
            }else{
              State[i]=IncomingByte;
            }
      }
      Entry[B] = getnum(&State[0]);
    }

    if((Entry[C] == Hist[C]) && (Entry[B] == Hist[B]))
      continue;
    for( ; TRUE; Past[C] = Entry[C], Past[B] = Entry[B]){
      if((Entry[C] == Past[C]) && (Entry[B] == Past[B]))
         break;

      /**Processing***/
      if(Entry[C] != Past[C]){
         if(!(Serial.available() > 0)){
           //leituras do microcontrolador.
           Serial.println(Entry[C],DEC);
           delay(10);
         }
      }

      if(Entry[B] != Past[B]){
           PORTB = Entry[B];
      }
    }
  }
}
//have to press reset in learning mode always.
/***FUNCTIONS***/
int getnum(char* x)
{
  int num;
  if(sscanf(x,"%d",&num) != 0){
    if (num == NULL)
      num = 0;
    return num;
  }else{
    return 0;
  }
}
//char *X, X is a variable that stores a fisical
//address with a cast type char.
//getnum works with string terminating with only NULL or '\0'
//my style
//Lesson always be flexible and except and obey the good practices,
// without any desobidience, and for extra culture if desired try to
// clarify the why things are as they are. Never be stuborn.
```

```
#include<avr/io.h>
#define TRUE 1
#define FALSE 0
#define BufSize 127
#define A 0
#define C 1

int getnum(char* x);

void setup()
{
 // start serial port at 9600 bps:
 Serial.begin(9600);
 DDRA=B00000000;
 PORTA=B11111111;
 DDRC=B11111111;
 PORTC=B00000000;
}

void loop()
{
 int i;
 int Entry[2];
 int Past[2];
 int Hist[2];
 char IncomingByte;
 char State[BufSize];


 Serial.flush();
 //INIC

 for( Past[A] = Entry[A], Past[C] = Entry[C]; TRUE; Hist[A] = Entry[A], Hist[C] = Entry[C]){
  //ENTRADA Portas
  Entry[A]=PINA;
  //ENTRADA Serial
  if(Serial.available() > 0){
   delay(25);//wait for incoming data.
   for(i = 0; IncomingByte = Serial.read(); i++){
      if((IncomingByte == '\r') || (IncomingByte == '\n')){
       State[i] = '\0';
       Serial.flush();
       break;
      }else{
       State[i]=IncomingByte;
      }
   }
```

```
      Entry[C] = getnum(&State[0]);
    }

    if((Entry[A] == Hist[A]) && (Entry[C] == Hist[C]))
      continue;
    for( ; TRUE; Past[A] = Entry[A], Past[C] = Entry[C]){
      if((Entry[A] == Past[A]) && (Entry[C] == Past[C]))
            break;

      /**Processing***/
      if(Entry[A] != Past[A]){
            if(!(Serial.available() > 0)){
              //leituras do microcontrolador.
              Serial.println(Entry[A],DEC);
              delay(10);
            }
      }

      if(Entry[C] != Past[C]){
              PORTC = Entry[C];
      }
    }
  }
}
//have to press reset in learning mode always.
/***FUNCTIONS***/
int getnum(char* x)
{
  int num;
  if(sscanf(x,"%d",&num) != 0){
    if (num == NULL)
      num = 0;
    return num;
  }else{
    return 0;
  }
}
//char* X, X is a variable that stores a fisical
//address with a cast type char.
//getnum works with string terminating with only NULL or '\0'
//my style
```

```
/***sergio manuel salazar dos santos***/
/***Tel:- 934805603***/
/***Rua do relogio 268, 4770-245 Joane, Vila Nova de famalicao, Braga, Portugal***/
/***CABECALHO Libraries***/
//this program is a learning finite state machine.
#include "creation_7.h"

/***MAIN***/
int main(int argc, char* argv[])
{

//capture prog arguments.
if(argc < 2){
	printf("Enter the board use duemilanove or mega or uno ? \n");
	return 0;
}else if(argc < 3){
	printf("Enter a file name please \n");
	return 0;
}else if(argc < 4){
	printf("Enter learning mode on or off \n");
	return 0;
}

printf("FILE -> %s\n", __FILE__);
printf("DATE -> %s TIME -> %s\n", __DATE__, __TIME__);
printf("DATE -> %d\n", __LINE__);
printf("double -> %d\n", Double(5));
//printf("Pow -> %f\n", Pow(2,5));
printf("argv[0] (Progname) -> %s	size: %d		\n", argv[0], strlen(argv[0]));
printf("argv[1] (Board)-> %s size: %d		\n", argv[1], strlen(argv[1]));
printf("argv[2] (Filename)-> %s	size: %d		\n", argv[2], strlen(argv[2]));
printf("argv[3] (Learnmode) -> %s	size: %d		\n", argv[3], strlen(argv[3]));

//make decisions relative income arguments.
if(strcmp(argv[1], "mega")==0){
	DEVICE = Putstr(DEVICE_1);
}else if(strcmp(argv[1], "duemilanove")==0){
	DEVICE = Putstr(DEVICE_2);
}else if(strcmp(argv[1], "uno")==0){
	DEVICE = Putstr(DEVICE_3);
}else{
	printf("Board options mega or duemilanove or uno\n");
	return 0;
}

if(strcmp(argv[3], "on")==0){
	LEARN=1;
}else if(strcmp(argv[3], "off")==0){
        LEARN=0;
}else{
        printf("LEARN mode options on or off\n");
        return 0;
}


/***Internal Variables***/
int errno;

/***portcomunication setup file descriptor***/
int fd;
int c, ires, ores;

fd = open(DEVICE, O_RDWR | O_NOCTTY | O_NDELAY);
printf("file descriptor: %d\n",fd);
if(fd < 0)
        goto EXIT_2;

//com prepare
struct termios oldtio,newtio;
tcgetattr(fd, &oldtio);
bzero(&newtio, sizeof(newtio));

/*
BAUDRATE: Set bps rate. You could also use cfsetispeed and cfsetospeed.
CRTSCTS : output hardware flow control (only used if the cable has
      all necessary lines. See sect. 7 of Serial-HOWTO)
CS8     : 8n1 (8bit,no parity,1 stopbit)
CLOCAL  : local connection, no modem control
CREAD   : enable receiving characters
*/
//newtio.c_cflag = BAUDRATE | CRTSCTS | CS8 | CLOCAL | CREAD;
newtio.c_cflag = BAUDRATE | CS8 | CLOCAL | CREAD;

/*
IGNPAR  : ignore bytes with parity errors
ICRNL   : map CR to NL (otherwise a CR input on the other computer
      will not terminate input)
otherwise make device raw (no other input processing)
*/
//newtio.c_iflag |= (IGNPAR | ICRNL);
newtio.c_iflag |= (INPCK | ISTRIP);
//newtio.c_iflag |= INPCK;

/*
Raw output. first option.
*/
```

```
//newtio.c_oflag |= 0;
//newtio.c_oflag |= (OPOST | ONLCR);
newtio.c_oflag |= OPOST;
//newtio.c_oflag &= ~OPOST;

/*
ICANON  : enable canonical input
disable all echo functionality, and don't send signals to calling program
*/
newtio.c_lflag = ICANON;

/*
initialize all control characters
default values can be found in /usr/include/termios.h, and are given
in the comments, but we don't need them here
*/
newtio.c_cc[VINTR]    = 0;     /* Ctrl-c */
newtio.c_cc[VQUIT]    = 0;     /* Ctrl-\ */
newtio.c_cc[VERASE]   = 0;     /* del */
newtio.c_cc[VKILL]    = 0;     /* @ */
newtio.c_cc[VEOF]     = 4;     /* Ctrl-d */
newtio.c_cc[VTIME]    = 0;     /* inter-character timer unused */
newtio.c_cc[VMIN]     = 1;     /* blocking read until 1 character arrives */
newtio.c_cc[VSWTC]    = 0;     /* '\0' */
newtio.c_cc[VSTART]   = 0;     /* Ctrl-q */
newtio.c_cc[VSTOP]    = 0;     /* Ctrl-s */
newtio.c_cc[VSUSP]    = 0;     /* Ctrl-z */
newtio.c_cc[VEOL]     = 0;     /* '\0' */
newtio.c_cc[VREPRINT] = 0;     /* Ctrl-r */
newtio.c_cc[VDISCARD] = 0;     /* Ctrl-u */
newtio.c_cc[VWERASE]  = 0;     /* Ctrl-w */
newtio.c_cc[VLNEXT]   = 0;     /* Ctrl-v */
newtio.c_cc[VEOL2]    = 0;     /* '\0' */

/*
now clean the modem line and activate the settings for the port
*/
tcflush(fd, TCIFLUSH);
if(tcsetattr(fd, TCSANOW, &newtio) != 0)
        goto EXIT_3;

/*******************************************/
/***nanosleep***/
//alivea procesador.
struct timespec* timer_1;
timer_1 = calloc(1, sizeof(struct timespec));
timer_1->tv_sec = 0;
timer_1->tv_nsec = 10000000;
struct timespec* timer_2;
```

```
timer_2 = calloc(1, sizeof(struct timespec));
timer_2->tv_sec = 1;
timer_2->tv_nsec = 0;
/******/
/***Variables***/
int i, j, n;
buildingx leitura;
buildingx memory;
pastx hist;
char ordem[SEND];

/***File that stores de structs***/
FILE* fsm;
fsm=fopen(argv[2], "a+");
if(fsm == NULL)
        goto EXIT_1;

//Inicialize the primary struct.
printf("struct size:%d\n\7", sizeof(leitura));
//saida ou estado inicial.

strcpy(&leitura.key[0][0], "\r\n");
strcpy(&leitura.key[1][0], "\r\n");
strcpy(&leitura.who[0][0], "\r\n");
strcpy(&leitura.who[1][0], "\r\n");

//delay.
nanosleep(timer_1, NULL);

printf("        Press reset button on target!!\n");
perror("status ");

//The main program starts here.
/***CICLOS MAQUINA***/
for(printf("WELCOME!!!\n"); TRUE; strcpy( &hist.registry[1][0], &leitura.key[1][0])){

        //ENTRADAS.
        //Entrada Serial
        ires=read(fd, &leitura.key[1][0], buf_size);

        //printf("%s", &leitura.key[1][0]);

        if(ires > 0){
                for(i=0; i < buf_size; i++){
                        if(leitura.key[1][i] == '\n'){
                                i++;
                                break;
                        }
                }
        leitura.key[1][i] = '\0';
    }else{
        //perror("status ");
        nanosleep(timer_1,NULL);
        continue;
    }

    if(strcmp( &leitura.key[1][0], &hist.registry[1][0])==0)
        continue;
    printf("\nleitura : %d ->  %s", ires, &leitura.key[1][0]);
    if(strcmp(&leitura.key[0][0],&leitura.key[1][0])==0)
        continue;
    /****************************************************************************
******************/

        //Search Engine
        for(i=0, rewind(fsm); fread(&memory, sizeof(buildingx), 1, fsm); i++){
            //found we can choose the depth of FSM here.
            if(
                strcmp(&leitura.key[0][0], &memory.key[0][0])==0 &&
                strcmp(&leitura.key[1][0], &memory.key[1][0])==0 &&
                strcmp(&leitura.who[0][0], &memory.who[0][0])==0    ){

                printf("struct = %d ", i);
                printf("out -> %s", &memory.who[1][0]);
                //preset
                strcpy(&leitura.key[0][0],&leitura.key[1][0]);
                strcpy(&leitura.who[0][0], &memory.who[1][0]);

                if(strcmp(&leitura.who[0][0], "quit\r\n")==0){
                    ores = write(fd, "0\r\n", 4*sizeof(char));
                    goto EXIT_1;
                }else{
                    //Get number string

                    if(sscanf(&leitura.who[0][0], "%[0-9]", ordem) == 0)
                        strcpy(ordem,"0");
                    strcat(ordem,"\r\n");

                    //printf("ordem -> %s", ordem);

                    ores = write(fd,ordem, strlen(ordem));
                    break;
                }
            }
            //error
            if(ferror(fsm)){
                perror("status ");
                errno = 0;
```

```
                                break;
                        }
                }

                //procedures executed only if in LEARN mode on.
                if(feof(fsm)){
                        if(LEARN){
                                printf("--> new data\n");
                                infor(&leitura.who[1][0], buf_size, stdin);

                                if(strcmp(&leitura.who[1][0], "exit\r\n") == 0){
                                        goto EXIT_1;
                                }else{
                                        //Get number string

                                        if(sscanf(&leitura.who[1][0], "%[0-9]", ordem) == 0)
                                                strcpy(ordem,"0");
                                        strcat(ordem,"\r\n");

                                        ores = write(fd, ordem, strlen(ordem));
                                }

                                fseek(fsm, 0, SEEK_END);
                                fwrite(&leitura, sizeof(buildingx), 1, fsm);
                                //preset
                                strcpy(&leitura.key[0][0], &leitura.key[1][0]);
                                strcpy(&leitura.who[0][0], &leitura.who[1][0]);

                                printf("done!\n");
                                continue;
                        }else{
                                printf("Unknown input!\n");
                        }
        }
/
******************************************************************************
***********/
}
//EXITS in various points of the program.
EXIT_1:
        nanosleep(timer_2,NULL);
        /* restore the old port settings */
        tcsetattr(fd, TCSANOW, &oldtio);
        close(fd);
        free(DEVICE);
        free(timer_1);
        free(timer_2);
        fclose(fsm);
        return 0;
```

```
EXIT_2:
        perror(DEVICE);
        free(DEVICE);
        exit(-1);
        return 0;
EXIT_3:
        /* restore the old port settings */
        tcsetattr(fd, TCSANOW, &oldtio);
        close(fd);
        free(DEVICE);
        return 0;

}//main

//new aproach and I prefer it.
//a thing is returning a pointer.
//another is writting to an address.
//beautifull work.
//pointer are easy but take work in allocating
//memory all the time. function ReadConsole and Putstr demonstrates just that.
//obra de arte.
//magic formula, address=pow(2,0)*input+pow(2,8)*output;
//Going to try to save the file inside the microcontroller so that it will
//run independently without Serial comunication.
//There is no such thing as future only coming present.
//search for more bugs.
//just for the fun of it going to try a new aproach and consider this stable and finished.
//New capabilities can be added later.
```

```
/*creation_7.h*/
#ifdef _CREATION_7_H_
#else
	#define _CREATION_7_H_
	//PROTOTYPE
	/**sergio manuel salazar dos santos 934805603**/
	/***Libraries***/
	// fopen perror fread fwrite feof fseek ferror fclose rewind scanf sscanf getchar scanf fscanf
	// strncpy sscanf
	#include <stdio.h>
	// calloc free realloc
	#include <stdlib.h>
	// strcpy strcmp strcat memcmp
	#include <string.h>
	// termios tcflush
	#include <termios.h>
	// nanosleep sleep
	#include <time.h>
	// tcflsuh read write close
	#include <unistd.h>
	// perror
	#include <errno.h>
	// open
	#include <sys/types.h>
	#include <sys/stat.h>
	#include <fcntl.h>
	//assert
	#include <assert.h>
	//offsetof
	#include <stddef.h>
	//__fpurge
	//#include <stdio_ext.h>
	//#include <netdb.h>
	//#include <netinet/in.h>
	//#include <stdbool.h>
	//#include <ctype.h>
	//#include <limits.h>
	//#include <semaphore.h>
	//#include <pthread.h>
	//#include <math.h>
	//#include <signal.h>
	//#include <sys/mman.h>
	//#include <sys/wait.h>
	//#include <sys/time.h>
	//#include <sys/resource.h>
	#include <sys/dir.h>
	//#include <sys/socket.h>
	//#include <sys/un.h>


	/***MACROS***/
	#define TRUE 1
	#define FALSE 0
	#define BAUDRATE B9600
	//arduino MEGA
	#define DEVICE_1 "/dev/serial/by-id/usb-FTDI_FT232R_USB_UART_A600aiS5-if00-port0"
	//arduino DUEMILANOVE
	#define DEVICE_2 "/dev/serial/by-id/usb-FTDI_FT232R_USB_UART_A9007Qwg-if00-
port0"
	//arduino uno
	#define DEVICE_3 "/dev/serial/by-id/usb-
Arduino__www.arduino.cc__Arduino_Uno_6493534313335121 21A0-if00"
	#define _POSIX_SOURCE 1
	#define buf_size 32
	#define SEND 16
	#define Double(x) (2*(x))

	/***Gloabal Variables***/
	int LEARN;
	char* DEVICE;

	/***PROTO***/
	typedef struct {
		char registry[2][buf_size];
	}pastx;

	typedef struct {
		char key[2][buf_size];
		char who[2][buf_size];
	}buildingx;

#endif

/***FUNCTION TITLES***/
#include "creation_7_func.h"
```

```
//PROTOTYPE
/**sergio manuel salazar dos santos 934805603**/

#include "creation_7.h"

/*FUNCOES*/
/*****Putstr*******/
char* Putstr(char* str)
{
	int i; char* ptr;
	ptr = (char*)calloc(strlen(str), sizeof(char));
	if(ptr == NULL){
		perror("NULL!\n");
		return NULL;
	}
	for(i=0; (ptr[i] = str[i]); i++)
	{
		if(ptr[i] == '\0')
			break;
	}
	return (ptr);
}

/***ReadConsole***/
char* ReadConsole(FILE* stream)
{
	int i, NBytes;
	char caracter;
	char* value=NULL;
	for(i=0, NBytes=8; (caracter=getc(stream)) != EOF; i++){
		if((i==NBytes) | (i==0)){
			NBytes=2*NBytes;
			value=(char*)realloc(value, NBytes*sizeof(char));
			if(value==NULL)
				perror(value);
		}
		*(value+i)=caracter;
		if(caracter=='\n'){
			*(value+i)='\0';
			break;
		}
	}
	return value;
}

/***getnum***/
int getnum(int min, int max)
```

```
{
	int num;
	for(num=0; (scanf("%d",&num)==0) || num<min || num>max ; getchar()){
		perror("loop status");
	}
	return num;
}

/*********fillvec************/
int* fillvec(int num, int size)
{
	int* x;
	int i;
	if(size > 0){
		x=calloc(size, sizeof(int));
		for(i=0; i<=size; i++)
		{
			x[i]=num;
			//printf("x[%d]-> %d\n",i,x[i]);//troubleshooting
		}
	}else{
		return NULL;
	}
	return x;
}

/*******ReadFiletoMem*******/
void* ReadFiletoMem(void* datatype, FILE* filename)
{
	int i , n;
	fseek(filename, 0, SEEK_SET);
	datatype=calloc(1,sizeof(*datatype));
	for(i=0, n=1; fread((datatype+i), sizeof(*datatype), 1, filename); datatype=realloc(datatype, 
n*sizeof(*datatype))){
		i++;
		n++;
		if(feof(filename))
			break;
		if(ferror(filename)){
			perror("status:");
			return NULL;
		}
	}
	return datatype;
}

/**GetChar()**/
unsigned char GetChar()
{
	unsigned char x;
	unsigned char value;
	for(value=getchar(); x!='\n'; x=getchar());
	if(value=='\0')
		return 1;
	x='0';
	return value;
}

/******/
//sintaxe muito muito importante, mais importante doque a semantica.
//I learn allot reading other peoples code.
char* ReadConsoleSer(FILE* stream)
{
	int i, NBytes;
	char caracter;
	char* value=NULL;
	for(i=0, NBytes=8; (caracter=getc(stream)) != EOF;i++){
		if((i==NBytes) | (i==0)){
			NBytes=2*NBytes;
			value=(char*)realloc(value,NBytes*sizeof(char)+2);
			if(value==NULL)
				perror(value);
		}
		*(value+i)=caracter;
		if(caracter=='\n'){
			*(value+i)='\r';
			i++;
			*(value+i)='\n';
			i++;
			*(value+i)='\0';
			break;
		}
	}
	return value;
}

/******/
int getnumber(char* x)
{
  int num;
  if(sscanf(x, "%d", &num)!=0)
    return num;
  else
    return 0;
}

/***infor***/
int infor(char* inf, int Size_Inf, FILE* stream){
	int n;
	char* a;
	a=calloc(1024, sizeof(char));
	while((n=fscanf(stream, "%s", a)) != 0){
		if(strlen(a) > (Size_Inf-3)){
			printf("overflow retry.\n");
			continue;
		}
		strncpy(inf, a, (Size_Inf-3));
		strcat(inf, "\r\n");
		break;
	}
	free(a);
	return n;
}

/******/
```

```
/*creation_7_func.h*/
#ifdef _CREATION_7_FUNC_H_
#else
        #define _CREATION_7_FUNC_H
        //PROTOTYPE
        /***FUNCTION TITLES***/
        /***Coloca uma string numa variavel tipo apontador allocando tamanho automatico***/
        char* Putstr(char* str);
        /***Lê stdin e aloca automaticamente espaço em memória***/
        char* ReadConsole();
        /***lê apenas numeros de uma string***/
        int getnum(int min, int max);
        /***preenche vector de tamanho size por num***/
        int* fillvec(int num, int size);
        /***ReadFiletoMem***/
        void* ReadFiletoMem(void* datatype,FILE* filename);
        /***GetChar***/
        unsigned char GetChar();
        /***ReadConsoleR***/
        char* ReadConsoleSer(FILE* stream);
        /***gets the number out of a string***/
        int getnumber(char* x);
        /*******/
        int infor(char* inf, int Size_Inf, FILE* stream);
        /*******/
#endif
```

```
CC=gcc
LIB=-L./

all:resultado

resultado:creation_7.o creation_7_func.o
	${CC} creation_7.o creation_7_func.o -Wall -lm -o FSM.exe ${LIB}

resultado.o:creation_7.c
	${CC} -c creation_7.c -Wall -lm -o creation_7.o ${LIB}

resultado_func.o:creation_7_func.c
	${CC} -c creation_7_func.c -Wall -lm -o creation_7_func.o ${LIB}

clean.o:
	rm *.o *.exe
```

```
/***sergio manuel salazar dos santos***/
/***Tel:- 934805603***/
/***Rua do relogio 268, 4770-245 Joane, Vila Nova de famalicao, Braga, Portugal***/
/***CABECALHO Libraries***/
//this program is a learning finite state machine.
#include "creation_7.h"

/***MAIN***/
int main(int argc, char* argv[])
{

if(1){
  printf("MAIN Header File Loaded.\n");
}
if(1){
  printf("FUNCTION Header File Loaded.\n");
}

//capture prog arguments.
if(argc < 2){
  printf("Enter the board use duemilanove or mega or uno or none ? \n");
  return 0;
}else if(argc < 3){
  printf("Enter a file name please \n");
  return 0;
}else if(argc < 4){
  printf("Enter learning mode on or off \n");
  return 0;
}

printf("FILE -> %s\n", __FILE__);
printf("DATE -> %s TIME -> %s\n", __DATE__, __TIME__);
printf("DATE -> %d\n", __LINE__);
printf("double -> %d\n", Double(5));
//printf("Pow -> %f\n", Pow(2,5));
printf("argv[0] (Progname) -> %s     size: %d         \n", argv[0], strlen(argv[0]));
printf("argv[1] (Board)-> %s size: %d          \n", argv[1], strlen(argv[1]));
printf("argv[2] (Filename)-> %s       size: %d         \n", argv[2], strlen(argv[2]));
printf("argv[3] (Learnmode) -> %s   size: %d         \n", argv[3], strlen(argv[3]));

//make decisions relative income arguments.
if(strcmp(argv[1], "mega")==0){
  DEVICE = Putstr(DEVICE_1);
}else if(strcmp(argv[1], "duemilanove")==0){
  DEVICE = Putstr(DEVICE_2);
}else if(strcmp(argv[1], "uno")==0){
  DEVICE = Putstr(DEVICE_3);
}else if(strcmp(argv[1], "none")==0){
  DEVICE = Putstr(DEVICE_4);
}else{
  printf("Board options mega or duemilanove or uno or none\n");
  return 0;
}

if(strcmp(argv[3], "on")==0){
  LEARN=1;
}else if(strcmp(argv[3], "off")==0){
  LEARN=0;
}else{
  printf("LEARN mode options on or off\n");
  return 0;
}


/***Internal Variables***/
int errno;

/***portcomunication setup file descriptor***/
int fd;
int c, ires, ores;

fd = open(DEVICE,10);
//O_RDWR | O_NOCTTY | O_NDELAY);
printf("file descriptor: %d\n",fd);
if(fd < 0)
  goto EXIT_2;


/**********
//com prepare for USB protocol
struct termios oldtio,newtio;
tcgetattr(fd, &oldtio);
bzero(&newtio, sizeof(newtio));


//BAUDRATE: Set bps rate. You could also use cfsetispeed and cfsetospeed.
//CRTSCTS : output hardware flow control (only used if the cable has
//          all necessary lines. See sect. 7 of Serial-HOWTO)
//CS8     : 8n1 (8bit,no parity,1 stopbit)
//CLOCAL  : local connection, no modem control
//CREAD   : enable receiving characters

//newtio.c_cflag = BAUDRATE | CRTSCTS | CS8 | CLOCAL | CREAD;
newtio.c_cflag = BAUDRATE | CS8 | CLOCAL | CREAD;
```

```
//IGNPAR  : ignore bytes with parity errors
//ICRNL   : map CR to NL (otherwise a CR input on the other computer
//          will not terminate input)
//otherwise make device raw (no other input processing)

//newtio.c_iflag |= (IGNPAR | ICRNL);
newtio.c_iflag |= (INPCK | ISTRIP);
//newtio.c_iflag |= INPCK;


//Raw output. first option.

//newtio.c_oflag |= 0;
//newtio.c_oflag |= (OPOST | ONLCR);
newtio.c_oflag |= OPOST;
//newtio.c_oflag &= ~OPOST;


//ICANON  : enable canonical input
//disable all echo functionality, and don't send signals to calling program

newtio.c_lflag = ICANON;


//initialize all control characters
//default values can be found in /usr/include/termios.h, and are given
//in the comments, but we don't need them here

newtio.c_cc[VINTR]    = 0;     // Ctrl-c
newtio.c_cc[VQUIT]    = 0;     // Ctrl-\
newtio.c_cc[VERASE]   = 0;     // del
newtio.c_cc[VKILL]    = 0;     // @
newtio.c_cc[VEOF]     = 4;     // Ctrl-d
newtio.c_cc[VTIME]    = 0;     // inter-character timer unused
newtio.c_cc[VMIN]     = 1;     // blocking read until 1 character arrives
newtio.c_cc[VSWTC]    = 0;     // '\0'
newtio.c_cc[VSTART]   = 0;     // Ctrl-q
newtio.c_cc[VSTOP]    = 0;     // Ctrl-s
newtio.c_cc[VSUSP]    = 0;     // Ctrl-z
newtio.c_cc[VEOL]     = 0;     // '\0'
newtio.c_cc[VREPRINT] = 0;     // Ctrl-r
newtio.c_cc[VDISCARD] = 0;     // Ctrl-u
newtio.c_cc[VWERASE]  = 0;     // Ctrl-w
newtio.c_cc[VLNEXT]   = 0;     // Ctrl-v
newtio.c_cc[VEOL2]    = 0;     // '\0'


//now clean the modem line and activate the settings for the port
```

```
tcflush(fd, TCIFLUSH);
if(tcsetattr(fd, TCSANOW, &newtio) != 0)
  goto EXIT_3;

*******************************************/
/***nanosleep***/
//alivea procesador.
struct timespec* timer_1;
timer_1 = calloc(1, sizeof(struct timespec));
timer_1->tv_sec = 0;
timer_1->tv_nsec = 10000000;
struct timespec* timer_2;
timer_2 = calloc(1, sizeof(struct timespec));
timer_2->tv_sec = 1;
timer_2->tv_nsec = 0;
/******/
/***Variables***/
int i, j, n;
buildingx leitura;
buildingx memory;
pastx hist;
char ordem[LINE];

/***File that stores de structs***/
FILE* fsm;
fsm=fopen(argv[2], "a+");
if(fsm == NULL)
  goto EXIT_1;
FILE* ftemp;
ftemp=fopen("temp.txt", "a+");
if(ftemp == NULL)
  goto EXIT_1;

//Inicialize the primary struct.
printf("struct size:%d\n\7", sizeof(leitura));
//saida ou estado inicial.

strcpy(&leitura.Input[0][0], "None");
strcpy(&leitura.Input[1][0], "None");
strcpy(&leitura.Output[0][0], "None");
strcpy(&leitura.Output[1][0], "None");

//delay.
nanosleep(timer_1, NULL);

printf("            Press reset button on target!!\n");
perror("status ");

//The main program starts here.
```

```
//fprintf(ftemp,"%s     %s       %s       %s\n","one","two","three","four");




/***CICLOS MAQUINA***/
for(printf("WELCOME!!!\n"); TRUE; strcpy( hist.registry[1], leitura.Input[1])){

  //ENTRADAS.
  //Entrada Serial
  //ires=read(fd, &leitura.Input[1][0], buf_size);
  ires=0;
  //fgets(&leitura.Input[1][0],buf_size,stdin);
  printf("Input ->          ");
  strcpy(leitura.Input[1],ReadConsole(stdin));
  //printf("Entry : %s.\n", &leitura.Input[1][0]);
  /***/

  printf("leitura : %d ->  %s\n", ires, leitura.Input[1]);

  if(strcmp(leitura.Input[1], "quit")==0)
    goto EXIT_1;

  /*******************************************************************************
**************/
  //very impotante in real time
  if(strcmp( leitura.Input[1], hist.registry[1])==0)
    continue;
  if(strcmp(leitura.Input[0],leitura.Input[1])==0)
    continue;
  /*******************************************************************************
**************/

  //Search Engine
  //fread(&memory, sizeof(buildingx), 1, fsm)
  printf("eter\n");
  for(i=0,rewind(ftemp);fscanf(ftemp,"%s     %s       %s       %s\n",
memory.Input[0],memory.Output[0],memory.Input[1],memory.Output[1])!=EOF; i++){

    //printf("memory : %s<>%s<>%s<->
%s\n",memory.Input[0],memory.Output[0],memory.Input[1],memory.Output[1]);
    //printf("leitura : %s<>%s<>%s<->
%s\n",leitura.Input[0],leitura.Output[0],leitura.Input[1],leitura.Output[1]);

    //found we can choose the depth of FSM here.
    if(
      strcmp(leitura.Input[0], memory.Input[0])==0
      &&
      strcmp(leitura.Input[1], memory.Input[1])==0
      &&
      strcmp(leitura.Output[0], memory.Output[0])==0
                                                          ){

      printf("line %d out ->                              %s\n", i, memory.Output[1]);
      //preset
      strcpy(leitura.Input[0], leitura.Input[1]);
      strcpy(leitura.Output[0], memory.Output[1]);

      strcpy(ordem,memory.Output[1]);
      strcat(ordem,"\r\n");
      ores = write(fd,ordem,strlen(ordem));
      break;
    }
    //error
    if(ferror(ftemp)){
      perror("status ");
      errno = 0;
      break;
    }
  }

  //procedures executed only if in LEARN mode on.
  if(feof(ftemp)){
    if(LEARN){
      printf("--> new data\n");
      strcpy(leitura.Output[1],ReadConsole(stdin));

      if(strcmp(leitura.Output[1],"quit") == 0){
       goto EXIT_1;
      }else{

       strcpy(ordem,leitura.Output[1]);
       strcat(ordem,"\r\n");
       ores = write(fd,ordem,strlen(ordem));
      }

      fseek(ftemp, 0, SEEK_END);
      fprintf(ftemp,"%s %s      %s      %s\n",
leitura.Input[0],leitura.Output[0],leitura.Input[1],leitura.Output[1]);

      //preset
      strcpy(leitura.Input[0], leitura.Input[1]);
      strcpy(leitura.Output[0], leitura.Output[1]);

      printf("done!\n");
      continue;
    }else{
      printf("Unknown or Repeated input!\n");
      printf("line %d out ->                                    %s\n", i, memory.Output[1]);
    }
  }
  /*******************************************************************************
**************/
}
//EXITS in various points of the program.
EXIT_1:
  nanosleep(timer_2,NULL);
  /* restore the old port settings */
  //tcsetattr(fd, TCSANOW, &oldtio);
  close(fd);
  free(DEVICE);
  free(timer_1);
  free(timer_2);
  fclose(fsm);
  fclose(ftemp);
  return 0;

EXIT_2:
  perror(DEVICE);
  free(DEVICE);
  exit(-1);
  return 0;
EXIT_3:
  /* restore the old port settings */

  //tcsetattr(fd, TCSANOW, &oldtio);
  close(fd);
  free(DEVICE);
  return 0;

}//main

//new aproach and I prefer it.
//a thing is returning a pointer.
//another is writting to an address.
//beautifull work.
//pointer are easy but take work in allocating
//memory all the time. function ReadConsole and Putstr demonstrates just that.
//obra de arte.
//magic formula, address=pow(2,0)*input+pow(2,8)*output;
//Going to try to save the file inside the microcontroller so that it will
//run independently without Serial comunication.
//There is no such thing as future only coming present.
//search for more bugs.
//just for the fun of it going to try a new aproach and consider this stable and finished.
//New capabilities can be added later.
```

```
/*creation_7.h*/
#ifdef _CREATION_7_H_
#else
  #define _CREATION_7_H_
  //PROTOTYPE
  /**sergio manuel salazar dos santos 934805603**/
  /***Libraries***/
  // fopen perror fread fwrite feof fseek ferror fclose rewind scanf sscanf getchar scanf fscanf
  // strncpy sscanf
  #include <stdio.h>
  // calloc free realloc
  #include <stdlib.h>
  // strcpy strcmp strcat memcmp
  #include <string.h>
  // termios tcflush
  #include <termios.h>
  // nanosleep sleep
  #include <time.h>
  // tcflsuh read write close
  #include <unistd.h>
  // perror
  #include <errno.h>
  // open
  #include <sys/types.h>
  #include <sys/stat.h>
  #include <fcntl.h>
  //assert
  #include <assert.h>
  //offsetof
  #include <stddef.h>
  //__fpurge
  //#include <stdio_ext.h>
  //#include <netdb.h>
  //#include <netinet/in.h>
  //#include <stdbool.h>
  //#include <ctype.h>
  //#include <limits.h>
  //#include <semaphore.h>
  //#include <pthread.h>
  //#include <math.h>
  //#include <signal.h>
  //#include <sys/mman.h>
  //#include <sys/wait.h>
  //#include <sys/time.h>
  //#include <sys/resource.h>
  #include <sys/dir.h>
  //#include <sys/socket.h>
  //#include <sys/un.h>


  /***MACROS***/
  #define TRUE 1
  #define FALSE 0
  #define BAUDRATE B9600
  //arduino MEGA
  #define DEVICE_1 "/dev/serial/by-id/usb-FTDI_FT232R_USB_UART_A600aiS5-if00-port0"
  //arduino DUEMILANOVE
  #define DEVICE_2 "/dev/serial/by-id/usb-FTDI_FT232R_USB_UART_A9007Qwg-if00-port0"
  //arduino uno
  #define DEVICE_3 "/dev/serial/by-id/usb-
Arduino__www.arduino.cc__Arduino_Uno_649353431333512121A0-if00"
  #define DEVICE_4 "Com.txt"
  #define _POSIX_SOURCE 1
  #define buf_size 32
  #define SEND 16
  #define LINE 80
  #define Double(x) (2*(x))

  /***Gloabal Variables***/
  int LEARN;
  char* DEVICE;

  /***PROTO***/
  typedef struct {
    char registry[2][buf_size];
  }pastx;

  typedef struct {
    char Input[2][buf_size];
    //char *key[2];
    char Output[2][buf_size];
  }buildingx;



#endif

/***FUNCTION TITLES***/
#include "creation_7_func.h"
```

```
//PROTOTYPE
/**sergio manuel salazar dos santos 934805603**/

#include "creation_7.h"

/*FUNCOES*/
/*****Putstr*******/
char* Putstr(char* str)
{
  int i; char* ptr;
  ptr = (char*)calloc(strlen(str), sizeof(char));
  if(ptr == NULL){
    perror("NULL!\n");
    return NULL;
  }
  for(i=0; (ptr[i] = str[i]); i++)
  {
    if(ptr[i] == '\0')
      break;
  }
  return (ptr);
}

/***ReadConsole***/
char* ReadConsole(FILE* stream)
{
  int i, NBytes;
  char caracter;
  char* value=NULL;
  for(i=0, NBytes=8; (caracter=getc(stream)) != EOF; i++){
    if((i==NBytes) | (i==0)){
      NBytes=2*NBytes;
      value=(char*)realloc(value, NBytes*sizeof(char));
      if(value==NULL)
        perror(value);
    }
    *(value+i)=caracter;
    if(caracter=='\n'){
      *(value+i)='\0';
      break;
    }
  }
  return value;
}

/***getnum***/
int getnum(int min, int max)
```

```
{
  int num;
  for(num=0; (scanf("%d",&num)==0) || num<min || num>max ; getchar()){
    perror("loop status");
  }
  return num;
}

/*********fillvec************/
int* fillvec(int num, int size)
{
  int* x;
  int i;
  if(size > 0){
    x=calloc(size, sizeof(int));
    for(i=0; i<=size; i++)
    {
      x[i]=num;
      //printf("x[%d]-> %d\n",i,x[i]);//troubleshooting
    }
  }else{
    return NULL;
  }
  return x;
}

/*******ReadFiletoMem*******/
void* ReadFiletoMem(void* datatype, FILE* filename)
{
  int i , n;
  fseek(filename, 0, SEEK_SET);
  datatype=calloc(1,sizeof(*datatype));
  for(i=0, n=1; fread((datatype+i), sizeof(*datatype), 1, filename); datatype=realloc(datatype,
n*sizeof(*datatype))){
    i++;
    n++;
    if(feof(filename))
      break;
    if(ferror(filename)){
      perror("status:");
      return NULL;
    }
  }
  return datatype;
}

/**GetChar()**/
unsigned char GetChar()
{
  unsigned char x;
  unsigned char value;
  for(value=getchar(); x!='\n'; x=getchar());
  if(value=='\0')
    return 1;
  x='0';
  return value;
}

/******/
//sintaxe muito muito importante, mais importante doque a semantica.
//I learn allot reading other peoples code.
char* ReadConsoleSer(FILE* stream)
{
  int i, NBytes;
  char caracter;
  char* value=NULL;
  for(i=0, NBytes=8; (caracter=getc(stream)) != EOF;i++){
    if((i==NBytes) | (i==0)){
      NBytes=2*NBytes;
      value=(char*)realloc(value,NBytes*sizeof(char)+2);
      if(value==NULL)
        perror(value);
    }
    *(value+i)=caracter;
    if(caracter=='\n'){
      *(value+i)='\r';
      i++;
      *(value+i)='\n';
      i++;
      *(value+i)='\0';
      break;
    }
  }
  return value;
}

/******/
int getnumber(char* x)
{
  int num;
  if(sscanf(x, "%d", &num)!=0)
    return num;
  else
    return 0;
}

/***infor***/
int infor(char* inf, int Size_Inf, FILE* stream){
  int n;
  char* a;
  a=calloc(1024, sizeof(char));
  while((n=fscanf(stream, "%s", a)) != 0){
    if(strlen(a) > (Size_Inf-3)){
      printf("overflow retry.\n");
      continue;
    }
    strncpy(inf, a, (Size_Inf-3));
    strcat(inf, "\r\n");
    break;
  }
  free(a);
  return n;
}

/******/
```

```
/*creation_7_func.h*/
#ifdef _CREATION_7_FUNC_H_
#else
  #define _CREATION_7_FUNC_H
  //PROTOTYPE
  /***FUNCTION TITLES***/
  /***Coloca uma string numa variavel tipo apontador allocando tamanho automatico***/
  char* Putstr(char* str);
  /***Lê stdin e aloca automaticamente espaço em memória***/
  char* ReadConsole();
  /***lê apenas numeros de uma string***/
  int getnum(int min, int max);
  /***preenche vector de tamanho size por num***/
  int* fillvec(int num, int size);
  /***ReadFiletoMem***/
  void* ReadFiletoMem(void* datatype,FILE* filename);
  /***GetChar***/
  unsigned char GetChar();
  /***ReadConsoleR***/
  char* ReadConsoleSer(FILE* stream);
  /***gets the number out of a string***/
  int getnumber(char* x);
  /*******/
  int infor(char* inf, int Size_Inf, FILE* stream);
  /*******/



#endif
```

```
CC=gcc
LIB=-L./

all:resultado

resultado:creation_7.o creation_7_func.o
	${CC} creation_7.o creation_7_func.o -Wall -lm -o FSM.exe ${LIB}

resultado.o:creation_7.c
	${CC} -c creation_7.c -Wall -lm -o creation_7.o ${LIB}

resultado_func.o:creation_7_func.c
	${CC} -c creation_7_func.c -Wall -lm -o creation_7_func.o ${LIB}

clean.o:
	rm *.o *.exe
```

```
//Este programa é aplicado para botoneiras start/stop,
//eplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemialnove.
#include<avr/io.h>
#define TRUE 1
#define FALSE 0
#define column 4
#define lines 52

void setup()
{
 // start serial port at 9600 bps:
 //Serial.begin(9600);
 DDRC = B00000000;
 PORTC = B11111111;
 DDRB = B11111111;
 PORTB = B00000000;
}

void loop()
{
 int i, l, c;
 int data[5];
 int Hist[5];
 int count;
 //inic key
 data[0] = 0;
 data[1] = 63;
 data[2] = 0;
 data[4] = 0;
 //mem prepared for depth 2 in FSM (finite state machine).
 const int mem[(lines+1)][(column+1)]=
 {
   { 0, 0, 1,300, 0},//timer 23
   { 0, 0, 0,21, 0},//1
   { 0, 0, 1,21, 1},//2
   { 0, 0, 0,17, 1},//3
   { 0, 0, 1,17, 1},//4
   { 0, 0, 0,20, 1},//5
   { 0, 0, 1,20, 1},//6
   { 0, 0, 0, 5, 1},//7
   { 0, 0, 1, 5, 1},//8
   { 0, 0, 1,29, 0},//9
   { 0, 0, 0,29, 0},//10
   { 0, 0, 1,23, 0},//11
   { 0, 0, 0,23, 0},//12
   { 0, 0, 1,31, 0},//13
```

```
  { 0, 0, 0,31, 0},//14
  { 0, 0, 1, 7, 0},//15
  { 0, 0, 0, 7, 0},//16
  { 0, 0, 1,13, 0},//17
  { 0, 0, 0,13, 0},//18
  { 0, 0, 1,37, 0},//19
  { 0, 0, 0,37, 0},//20
  { 0, 0, 1,28, 0},//21
  { 0, 0, 0,28, 0},//22
  { 0, 0, 1,25, 0},//23
  { 0, 0, 0,25, 0},//24
  { 0, 0, 1,17, 1},//25
  { 0, 0, 0,13, 0},//26
  { 0, 0, 0,53, 0},//27
  { 0, 0, 1,53, 0},//28
  { 0, 0, 0,53, 0},//29
  { 0, 0, 1,22, 0},//30
  { 0, 0, 0,22, 0},//31
  { 0, 0, 1,61, 0},//32
  { 0, 0, 0,61, 0},//33
  { 0, 0, 1,55, 0},//34
  { 0, 0, 0,55, 0},//35
  { 0, 0, 1,63, 0},//36
  { 0, 0, 0,63, 0},//37
  { 0, 0, 1,52, 0},//38
  { 0, 0, 0,52, 0},//39
  { 0, 0, 1,19, 0},//40
  { 0, 0, 0,19, 0},//41
  { 0, 0, 1, 9, 0},//42
  { 0, 0, 0, 9, 0},//43
  { 0, 0, 1,12, 0},//44
  { 0, 0, 0,12, 0},//45
  { 0, 0, 0,24, 0},//46
  { 0, 0, 0,24, 0},//47
  { 0, 0, 1,49, 0},//48
  { 0, 0, 0,49, 0}//49
};

/*******************CICLOS DE MAQUINA*********************/
for( Hist[0] = data[0], Hist[1] = data[1], Hist[2] = data[2], count = 0; TRUE; Hist[3] = data[3]){

  data[3] = PINC;

  delay(60);

  //if(!(Serial.available()>0)){
    //Serial.print(data[3]);
    //Serial.print(" , ");
    //Serial.println(count);
  //}

  //timer setup
  if(data[4] == 1){
    count++;
  }else{
    count = 0;
  }
  //catch timer
  if(count == 15000){
    data[3] = 300;
    count = 0;
  }
  //end timer

  /*********************/
  if(data[3] == Hist[3])
    continue;
  if(data[1] == data[3])
    continue;
  /*********************/
  /*********************Search and apply changes********************/
  for( l = 0, c = 0; l < lines   ; l++){
    if(c >= column){
      l--;
      data[4] = mem[l][c];
      PORTB = data[4];
      //update
      data[0] = data[2];
      data[1] = data[3];
      data[2] = data[4];
      Hist[0] = data[0];
      Hist[1] = data[1];
      Hist[2] = data[2];
      Hist[4] = data[4];
      break;
    }
    //startup c=2, c=0, for more precise.
    for(c=2; c < column; c++){
      if(data[c] == mem[l][c]){
        continue;
      }else{
        break;
      }
    }
  }
  /************************FALL THREW************************************/
```

```
   /*************************************************************************/
  }
}
//Nao existe futuro apenas present proximo.
//The more in depth the state machine is the more obidiente and dumb it becomes, because
//it will have to specify all possible cases.
```

```
//Este programa é aplicado para botoneiras start/stop,
//eplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemialnove.
#include<avr/io.h>
#define TRUE 1
#define FALSE 0
#define column 4
#define lines 52

void setup()
{
 // start serial port at 9600 bps:
 //Serial.begin(9600);
 DDRC = B00000000;
 PORTC = B11111111;
 DDRB = B11111111;
 PORTB = B00000000;
}

void loop()
{
 int i, l, c;
 int data[5];
 int Hist[5];
 int count;
 //inic key
 data[0] = 0;
 data[1] = 63;
 data[2] = 0;
 data[4] = 0;
 //mem prepared for depth 2 in FSM (finite state machine).
 const int mem[(lines+1)][(column+1)]=
 {
   { 0, 0, 1,300, 0},//timer 23
   { 0, 0, 0,21, 0},//1
   { 0, 0, 1,21, 1},//2
   { 0, 0, 0,17, 1},//3
   { 0, 0, 1,17, 1},//4
   { 0, 0, 0,20, 1},//5
   { 0, 0, 1,20, 1},//6
   { 0, 0, 0, 5, 1},//7
   { 0, 0, 1, 5, 1},//8
   { 0, 0, 1,29, 0},//9
   { 0, 0, 0,29, 0},//10
   { 0, 0, 1,23, 0},//11
   { 0, 0, 0,23, 0},//12
   { 0, 0, 1,31, 0},//13
```

```
  { 0, 0, 0,31, 0},//14
  { 0, 0, 1, 7, 0},//15
  { 0, 0, 0, 7, 0},//16
  { 0, 0, 1,13, 0},//17
  { 0, 0, 0,13, 0},//18
  { 0, 0, 1,37, 0},//19
  { 0, 0, 0,37, 0},//20
  { 0, 0, 1,28, 0},//21
  { 0, 0, 0,28, 0},//22
  { 0, 0, 1,25, 0},//23
  { 0, 0, 0,25, 0},//24
  { 0, 0, 1,17, 1},//25
  { 0, 0, 0,13, 0},//26
  { 0, 0, 0,53, 0},//27
  { 0, 0, 1,53, 0},//28
  { 0, 0, 0,53, 0},//29
  { 0, 0, 1,22, 0},//30
  { 0, 0, 0,22, 0},//31
  { 0, 0, 1,61, 0},//32
  { 0, 0, 0,61, 0},//33
  { 0, 0, 1,55, 0},//34
  { 0, 0, 0,55, 0},//35
  { 0, 0, 1,63, 0},//36
  { 0, 0, 0,63, 0},//37
  { 0, 0, 1,52, 0},//38
  { 0, 0, 0,52, 0},//39
  { 0, 0, 1,19, 0},//40
  { 0, 0, 0,19, 0},//41
  { 0, 0, 1, 9, 0},//42
  { 0, 0, 0, 9, 0},//43
  { 0, 0, 1,12, 0},//44
  { 0, 0, 0,12, 0},//45
  { 0, 0, 0,24, 0},//46
  { 0, 0, 0,24, 0},//47
  { 0, 0, 1,49, 0},//48
  { 0, 0, 0,49, 0}//49
};

/*******************CICLOS DE MAQUINA*********************/
for( Hist[0] = data[0], Hist[1] = data[1], Hist[2] = data[2], count = 0; TRUE; Hist[3] = data[3]){

  data[3] = PINC;

  delay(60);

  //if(!(Serial.available()>0)){
    //Serial.print(data[3]);
    //Serial.print(" , ");
    //Serial.println(count);
  //}

  //timer setup
  if(data[4] == 1){
    count++;
  }else{
    count = 0;
  }
  //catch timer
  if(count == 15000){
    data[3] = 300;
    count = 0;
  }
  //end timer

  /**********************/
  if(data[3] == Hist[3])
    continue;
  if(data[1] == data[3])
    continue;
  /*********************/
  /*********************Search and apply changes********************/
  for( l = 0, c = 0; l < lines   ; l++){
    if(c >= column){
      l--;
      data[4] = mem[l][c];
      PORTB = data[4];
      //update
      data[0] = data[2];
      data[1] = data[3];
      data[2] = data[4];
      Hist[0] = data[0];
      Hist[1] = data[1];
      Hist[2] = data[2];
      Hist[4] = data[4];
      break;
    }
    //startup c=2, c=0, for more precise.
    for(c=2; c < column; c++){
      if(data[c] == mem[l][c]){
        continue;
      }else{
        break;
      }
    }
  }
  /************************FALL THREW************************************/
```

```
   /************************************************************************/
  }
}
//Nao existe futuro apenas present proximo.
//The more in depth the state machine is the more obidiente and dumb it becomes, because
//it will have to specify all possible cases.
```

```
//Este programa é aplicado para botoneiras start/stop,
//aplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemialnove.
#include<avr/io.h>
#define TRUE 1
#define FALSE 0
#define column 4
#define lines 52

void setup()
{
 // start serial port at 9600 bps:
 //Serial.begin(9600);
 DDRC = B00000000;
 PORTC = B11111111;
 DDRB = B11111111;
 PORTB = B00000000;
}

void loop()
{
 int i, l, c;
 int data[5];
 int Hist[5];
 int count;
 //inic key
 data[0] = 0;
 data[1] = 63;
 data[2] = 0;
 data[4] = 0;
 //mem prepared for depth 2 in FSM (finite state machine).
 const int mem[(lines+1)][(column+1)]=
 {
   { 0, 0, 1,300, 0},//timer 23
   { 0, 0, 0,21, 0},//1
   { 0, 0, 1,21, 1},//2
   { 0, 0, 0,17, 1},//3
   { 0, 0, 1,17, 1},//4
   { 0, 0, 0,20, 1},//5
   { 0, 0, 1,20, 1},//6
   { 0, 0, 0, 5, 1},//7
   { 0, 0, 1, 5, 1},//8
   { 0, 0, 1,29, 0},//9
   { 0, 0, 0,29, 0},//10
   { 0, 0, 1,23, 0},//11
   { 0, 0, 0,23, 0},//12
   { 0, 0, 1,31, 0},//13
```

```
  { 0, 0, 0,31, 0},//14
  { 0, 0, 1, 7, 0},//15
  { 0, 0, 0, 7, 0},//16
  { 0, 0, 1,13, 0},//17
  { 0, 0, 0,13, 0},//18
  { 0, 0, 1,37, 0},//19
  { 0, 0, 0,37, 0},//20
  { 0, 0, 1,28, 0},//21
  { 0, 0, 0,28, 0},//22
  { 0, 0, 1,25, 0},//23
  { 0, 0, 0,25, 0},//24
  { 0, 0, 1,17, 1},//25
  { 0, 0, 0,13, 0},//26
  { 0, 0, 0,53, 0},//27
  { 0, 0, 1,53, 0},//28
  { 0, 0, 0,53, 0},//29
  { 0, 0, 1,22, 0},//30
  { 0, 0, 0,22, 0},//31
  { 0, 0, 1,61, 0},//32
  { 0, 0, 0,61, 0},//33
  { 0, 0, 1,55, 0},//34
  { 0, 0, 0,55, 0},//35
  { 0, 0, 1,63, 0},//36
  { 0, 0, 0,63, 0},//37
  { 0, 0, 1,52, 0},//38
  { 0, 0, 0,52, 0},//39
  { 0, 0, 1,19, 0},//40
  { 0, 0, 0,19, 0},//41
  { 0, 0, 1, 9, 0},//42
  { 0, 0, 0, 9, 0},//43
  { 0, 0, 1,12, 0},//44
  { 0, 0, 0,12, 0},//45
  { 0, 0, 0,24, 0},//46
  { 0, 0, 0,24, 0},//47
  { 0, 0, 1,49, 0},//48
  { 0, 0, 0,49, 0}//49
};

/*******************CICLOS DE MAQUINA*********************/
for( Hist[0] = data[0], Hist[1] = data[1], Hist[2] = data[2], count = 0; TRUE; Hist[3] = data[3]){

  data[3] = PINC;

  delay(60);

  //if(!(Serial.available()>0)){
    //Serial.print(data[3]);
    //Serial.print(" , ");
    //Serial.println(count);
  //}

  //timer setup
  if(data[4] == 1){
    count++;
  }else{
    count = 0;
  }
  //catch timer
  if(count == 15000){
    data[3] = 300;
    count = 0;
  }
  //end timer

  /*********************/
  if(data[3] == Hist[3])
    continue;
  if(data[1] == data[3])
    continue;
  /*********************/
  /*********************Search and apply changes********************/
  for( l = 0, c = 0; l < lines   ; l++){
    if(c >= column){
      l--;
      data[4] = mem[l][c];
      //update
      data[0] = data[2];
      data[1] = data[3];
      data[2] = data[4];
      Hist[0] = data[0];
      Hist[1] = data[1];
      Hist[2] = data[2];
      Hist[4] = data[4];
      //send
      PORTB = 63 & data[4];
      break;
    }
    //startup c=2, c=0, for more precise.
    for(c=2; c < column; c++){
      if(data[c] == mem[l][c]){
        continue;
      }else{
        break;
      }
    }
  }
  /************************FALL THREW**************************************/
```

```
    /************************************************************************/
  }
}
//Nao existe futuro apenas present proximo.
//The more in depth the state machine is the more obidiente and dumb it becomes, because
//it will have to specify all possible cases.
```

```
//Este programa é aplicado para botoneiras start/stop,
//aplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemialnove.
#include<avr/io.h>
#define TRUE 1
#define FALSE 0
#define column 4
#define lines 52

void setup()
{
 // start serial port at 9600 bps:
 //Serial.begin(9600);
 DDRC = B00000000;
 PORTC = B11111111;
 DDRB = B11111111;
 PORTB = B00000000;
}

void loop()
{
 int i, l, c;
 int data[5];
 int Hist[5];
 int count;
 //inic key
 data[0] = 0;
 data[1] = 63;
 data[2] = 0;
 data[4] = 0;
 //mem prepared for depth 2 in FSM (finite state machine).
 const int mem[(lines+1)][(column+1)]=
 {
   { 0, 0,65,300, 0},//timer 23
   { 0, 0, 0,21, 0},//1
   { 0, 0,65,21,65},//2
   { 0, 0, 0,17,65},//3
   { 0, 0,65,17,65},//4
   { 0, 0, 0,20,65},//5
   { 0, 0,65,20,65},//6
   { 0, 0, 0, 5,65},//7
   { 0, 0,65, 5,65},//8
   { 0, 0,65,29, 0},//9
   { 0, 0, 0,29, 0},//10
   { 0, 0,65,23, 0},//11
   { 0, 0, 0,23, 0},//12
   { 0, 0,65,31, 0},//13
```

```
  { 0, 0, 0,31, 0},//14
  { 0, 0,65, 7, 0},//15
  { 0, 0, 0, 7, 0},//16
  { 0, 0,65,13, 0},//17
  { 0, 0, 0,13, 0},//18
  { 0, 0,65,37, 0},//19
  { 0, 0, 0,37, 0},//20
  { 0, 0,65,28, 0},//21
  { 0, 0, 0,28, 0},//22
  { 0, 0,65,25, 0},//23
  { 0, 0, 0,25, 0},//24
  { 0, 0,65,17,65},//25
  { 0, 0, 0,13, 0},//26
  { 0, 0, 0,53, 0},//27
  { 0, 0,65,53, 0},//28
  { 0, 0, 0,53, 0},//29
  { 0, 0,65,22, 0},//30
  { 0, 0, 0,22, 0},//31
  { 0, 0,65,61, 0},//32
  { 0, 0, 0,61, 0},//33
  { 0, 0,65,55, 0},//34
  { 0, 0, 0,55, 0},//35
  { 0, 0,65,63, 0},//36
  { 0, 0, 0,63, 0},//37
  { 0, 0,65,52, 0},//38
  { 0, 0, 0,52, 0},//39
  { 0, 0,65,19, 0},//40
  { 0, 0, 0,19, 0},//41
  { 0, 0,65, 9, 0},//42
  { 0, 0, 0, 9, 0},//43
  { 0, 0,65,12, 0},//44
  { 0, 0, 0,12, 0},//45
  { 0, 0, 0,24, 0},//46
  { 0, 0, 0,24, 0},//47
  { 0, 0,65,49, 0},//48
  { 0, 0, 0,49, 0}//49
};

/*******************CICLOS DE MAQUINA*********************/
for( Hist[0] = data[0], Hist[1] = data[1], Hist[2] = data[2], count = 0; TRUE; Hist[3] = data[3]){

  data[3] = PINC;

  delay(60);

  //if(!(Serial.available()>0)){
    //Serial.print(data[3]);
    //Serial.print(" , ");
    //Serial.println(count);
  //}

  //timer setup
  if(data[4] == 65){
    count++;
  }else{
    count = 0;
  }
  //catch timer
  if(count == 15000){
    PORTB = 0;
    count = 0;
  }
  //end timer

  /*********************/
  if(data[3] == Hist[3])
    continue;
  if(data[1] == data[3])
    continue;
  /*********************/
  /*********************Search and apply changes********************/
  for( l = 0, c = 0; l < lines   ; l++){
    if(c >= column){
      l--;
      data[4] = mem[l][c];
      //update
      data[0] = data[2];
      data[1] = data[3];
      data[2] = data[4];
      Hist[0] = data[0];
      Hist[1] = data[1];
      Hist[2] = data[2];
      Hist[4] = data[4];
      //send
      PORTB = 63 & data[4];
      break;
    }
    //startup c=2, c=0, for more precise.
    for(c=2; c < column; c++){
      if(data[c] == mem[l][c]){
        continue;
      }else{
        break;
      }
    }
  }
  /************************FALL THREW*************************************/
```

```
    /****************************************************************************/
  }
}
//Nao existe futuro apenas present proximo.
//The more in depth the state machine is the more obidiente and dumb it becomes, because
//it will have to specify all possible cases.
```

```
//Este programa é aplicado para botoneiras start/stop,
//aplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemialnove.
#include<avr/io.h>
#include<avr/interrupt.h>
#define TRUE 1
#define FALSE 0
#define column 4
#define lines 52

void setup()
{
  // start serial port at 9600 bps:
  //Serial.begin(9600);
  DDRC = B00000000;
  PORTC = B11111111;
  DDRB = B11111111;
  PORTB = B00000000;
}

void loop()
{
  int i, l, c;
  int data[5];
  int Hist[5];
  int count=0;
  //inic key
  data[0] = 0;
  data[1] = 63;
  data[2] = 0;
  data[4] = 0;
  //mem prepared for depth 2 in FSM (finite state machine).
  const int mem[(lines+1)][(column+1)]=
  {
    { 0, 0,65,300, 0},//timer off
    { 0, 0, 0,21, 0},//1
    { 0, 0,65,21,65},//2
    { 0, 0, 0,17,65},//3timer on and output 1
    { 0, 0,65,17,65},//4
    { 0, 0, 0,20,65},//5
    { 0, 0,65,20,65},//6
    { 0, 0, 0, 5,65},//7
    { 0, 0,65, 5,65},//8
    { 0, 0,65,29, 0},//9
    { 0, 0, 0,29, 0},//10
    { 0, 0,65,23, 0},//11
    { 0, 0, 0,23, 0},//12
    { 0, 0,65,31, 0},//13
    { 0, 0, 0,31, 0},//14
    { 0, 0,65, 7, 0},//15
    { 0, 0, 0, 7, 0},//16
    { 0, 0,65,13, 0},//17
    { 0, 0, 0,13, 0},//18
    { 0, 0,65,37, 0},//19
    { 0, 0, 0,37, 0},//20
    { 0, 0,65,28, 0},//21
    { 0, 0, 0,28, 0},//22
    { 0, 0,65,25, 0},//23
    { 0, 0, 0,25, 0},//24
    { 0, 0,65,17,65},//25
    { 0, 0, 0,13, 0},//26
    { 0, 0, 0,53, 0},//27
    { 0, 0,65,53, 0},//28
    { 0, 0, 0,53, 0},//29
    { 0, 0,65,22, 0},//30
    { 0, 0, 0,22, 0},//31
    { 0, 0,65,61, 0},//32
    { 0, 0, 0,61, 0},//33
    { 0, 0,65,55, 0},//34
    { 0, 0, 0,55, 0},//35
    { 0, 0,65,63, 0},//36
    { 0, 0, 0,63, 0},//37
    { 0, 0,65,52, 0},//38
    { 0, 0, 0,52, 0},//39
    { 0, 0,65,19, 0},//40
    { 0, 0, 0,19, 0},//41
    { 0, 0,65, 9, 0},//42
    { 0, 0, 0, 9, 0},//43
    { 0, 0,65,12, 0},//44
    { 0, 0, 0,12, 0},//45
    { 0, 0, 0,24, 0},//46
    { 0, 0, 0,24, 0},//47
    { 0, 0,65,49, 0},//48
    { 0, 0, 0,49, 0}//49
  };

  /*******************CICLOS DE MAQUINA*********************/
  for( Hist[0] = data[0], Hist[1] = data[1], Hist[2] = data[2], count = 0; TRUE; Hist[3] = data[3]){

    data[3] = 63 & PINC;

    delay(60);

    //if(!(Serial.available()>0)){
      //Serial.print(data[3]);
      //Serial.print(" , ");
      //Serial.println(count);
    //}

    //timer setup
    if(data[4] == 65){
      count++;
    }else{
      count = 0;
    }
    //catch timer
    if(count == 15000){
      data[3] = 300;
      count = 0;
    }
    //end timer

    /**********************/
    if(data[3] == Hist[3])
      continue;
    if(data[1] == data[3])
      continue;
    /*********************/
    /*********************Search and apply changes********************/
    for( l = 0, c = 0; l < lines   ; l++){
      if(c >= column){
        l--;
        data[4] = mem[l][c];
        //update
        data[0] = data[2];
        data[1] = data[3];
        data[2] = data[4];
        Hist[0] = data[0];
        Hist[1] = data[1];
        Hist[2] = data[2];
        Hist[4] = data[4];
        //send
        PORTB = 63 & data[4];
        break;
      }
      //startup c=2, c=0, for more precise.
      for(c=2; c < column; c++){
        if(data[c] == mem[l][c]){
          continue;
        }else{
          break;
        }
      }
    }
    /************************FALL THREW*************************************/
```

```
    /***************************************************************************/
  }
}
//Nao existe futuro apenas present proximo.
//The more in depth the state machine is the more obidiente and dumb it becomes, because
//it will have to specify all possible cases.
```

```
//Este programa é aplicado para botoneiras start/stop,
//aplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemialnove.
#include<avr/io.h>
#include<avr/interrupt.h>
#define TRUE 1
#define FALSE 0
#define column 4
#define lines 52

void setup()
{
  // start serial port at 9600 bps:
  //Serial.begin(9600);
  DDRC = B00000000;
  PORTC = B11111111;
  DDRB = B11111111;
  PORTB = B00000000;
}

void loop()
{
  int i, l, c;
  int data[5];
  int Hist[5];
  int treat;
  int count=0;
  //inic key
  data[0] = 0;
  data[1] = 63;
  data[2] = 0;
  data[4] = 0;
  //mem prepared for depth 2 in FSM (finite state machine).
  const int mem[(lines+1)][(column+1)]=
  {
    { 0, 0,65,300, 0},//timer off output 0
    { 0, 0, 0,21, 0},//1
    { 0, 0,65,21,65},//2
    { 0, 0, 0,17,65},//3timer on and output 1
    { 0, 0,65,17,65},//4
    { 0, 0, 0,20,65},//5
    { 0, 0,65,20,65},//6
    { 0, 0, 0, 5,65},//7
    { 0, 0,65, 5,65},//8
    { 0, 0,65,29, 0},//9
    { 0, 0, 0,29, 0},//10
    { 0, 0,65,23, 0},//11
    { 0, 0, 0,23, 0},//12
    { 0, 0,65,31, 0},//13
    { 0, 0, 0,31, 0},//14
    { 0, 0,65, 7, 0},//15
    { 0, 0, 0, 7, 0},//16
    { 0, 0,65,13, 0},//17
    { 0, 0, 0,13, 0},//18
    { 0, 0,65,37, 0},//19
    { 0, 0, 0,37, 0},//20
    { 0, 0,65,28, 0},//21
    { 0, 0, 0,28, 0},//22
    { 0, 0,65,25, 0},//23
    { 0, 0, 0,25, 0},//24
    { 0, 0,65,17,65},//25
    { 0, 0, 0,13, 0},//26
    { 0, 0, 0,53, 0},//27
    { 0, 0,65,53, 0},//28
    { 0, 0, 0,53, 0},//29
    { 0, 0,65,22, 0},//30
    { 0, 0, 0,22, 0},//31
    { 0, 0,65,61, 0},//32
    { 0, 0, 0,61, 0},//33
    { 0, 0,65,55, 0},//34
    { 0, 0, 0,55, 0},//35
    { 0, 0,65,63, 0},//36
    { 0, 0, 0,63, 0},//37
    { 0, 0,65,52, 0},//38
    { 0, 0, 0,52, 0},//39
    { 0, 0,65,19, 0},//40
    { 0, 0, 0,19, 0},//41
    { 0, 0,65, 9, 0},//42
    { 0, 0, 0, 9, 0},//43
    { 0, 0,65,12, 0},//44
    { 0, 0, 0,12, 0},//45
    { 0, 0, 0,24, 0},//46
    { 0, 0, 0,24, 0},//47
    { 0, 0,65,49, 0},//48
    { 0, 0, 0,49, 0}//49
  };

  /*******************CICLOS DE MAQUINA*********************/
  for( Hist[0] = data[0], Hist[1] = data[1], Hist[2] = data[2], count = 0; TRUE; Hist[3] = data[3]){

    treat = 63 & PINC;
    data[3] = treat;

    delay(60);

    //if(!(Serial.available()>0)){
      //Serial.print(data[3]);
      //Serial.print(" , ");
      //Serial.println(count);
    //}

    //timer setup
    if(data[4] == 65){
      count++;
    }else{
      count = 0;
    }
    //catch timer
    if(count == 15000){
      data[3] = 300;
      count = 0;
    }
    //end timer

    /*********************/
    if(data[3] == Hist[3])
      continue;
    //if(!(Serial.available()>0)){
      //Serial.println(data[3]);
    //}
//    if(data[1] == data[3])
//      continue;
    /*********************/
    /*********************Search and apply changes********************/
    for( l = 0, c = 0; l < lines   ; l++){
      if(c >= column){
        l--;
        data[4] = mem[l][c];
        //update
        data[0] = data[2];
        data[1] = data[3];
        data[2] = data[4];
        Hist[0] = data[0];
        Hist[1] = data[1];
        Hist[2] = data[2];
        Hist[4] = data[4];
        //send
        PORTB = 63 & data[4];
        break;
      }
      //startup c=2, c=0, for more precise.
      for(c=2; c < column; c++){
        if(data[c] == mem[l][c]){
          continue;
        }else{
```

```
        break;
      }
    }
  }
  /**************************FALL THREW***************************************/




  /**********************************************************************************/
 }
}
//Nao existe futuro apenas present proximo.
//The more in depth the state machine is the more obidiente and dumb it becomes, because
//it will have to specify all possible cases.
```

```
//Este programa é aplicado para botoneiras start/stop,
//aplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemialnove.
#include<avr/io.h>
#include<avr/interrupt.h>
#define TRUE 1
#define FALSE 0
#define column 4
#define lines 52

void setup()
{
  // start serial port at 9600 bps:
  //Serial.begin(9600);
  DDRC = B00000000;
  PORTC = B11111111;
  DDRB = B11111111;
  PORTB = B00000000;
}

void loop()
{
  int i, l, c;
  int data[5];
  int Hist[5];
  int treat;
  int count=0;
  //inic key
  data[0] = 0;
  data[1] = 63;
  data[2] = 0;
  data[4] = 0;
  //mem prepared for depth 2 in FSM (finite state machine).
  const int mem[(lines+1)][(column+1)]=
  {
    { 0, 0,65,300, 0},//timer off output 0
    { 0, 0, 0,21, 0},//1
    { 0, 0,65,21,65},//2
    { 0, 0, 0,17,65},//3timer on and output 1
    { 0, 0,65,17,65},//4
    { 0, 0, 0,20,65},//5
    { 0, 0,65,20,65},//6
    { 0, 0, 0, 5,65},//7
    { 0, 0,65, 5,65},//8
    { 0, 0,65,29, 0},//9
    { 0, 0, 0,29, 0},//10
    { 0, 0,65,23, 0},//11
    { 0, 0, 0,23, 0},//12
    { 0, 0,65,31, 0},//13
    { 0, 0, 0,31, 0},//14
    { 0, 0,65, 7, 0},//15
    { 0, 0, 0, 7, 0},//16
    { 0, 0,65,13, 0},//17
    { 0, 0, 0,13, 0},//18
    { 0, 0,65,37, 0},//19
    { 0, 0, 0,37, 0},//20
    { 0, 0,65,28, 0},//21
    { 0, 0, 0,28, 0},//22
    { 0, 0,65,25, 0},//23
    { 0, 0, 0,25, 0},//24
    { 0, 0,65,17,65},//25
    { 0, 0, 0,13, 0},//26
    { 0, 0, 0,53, 0},//27
    { 0, 0,65,53, 0},//28
    { 0, 0, 0,53, 0},//29
    { 0, 0,65,22, 0},//30
    { 0, 0, 0,22, 0},//31
    { 0, 0,65,61, 0},//32
    { 0, 0, 0,61, 0},//33
    { 0, 0,65,55, 0},//34
    { 0, 0, 0,55, 0},//35
    { 0, 0,65,63, 0},//36
    { 0, 0, 0,63, 0},//37
    { 0, 0,65,52, 0},//38
    { 0, 0, 0,52, 0},//39
    { 0, 0,65,19, 0},//40
    { 0, 0, 0,19, 0},//41
    { 0, 0,65, 9, 0},//42
    { 0, 0, 0, 9, 0},//43
    { 0, 0,65,12, 0},//44
    { 0, 0, 0,12, 0},//45
    { 0, 0, 0,24, 0},//46
    { 0, 0, 0,24, 0},//47
    { 0, 0,65,49, 0},//48
    { 0, 0, 0,49, 0}//49
  };

  /******************CICLOS DE MAQUINA*********************/
  for( Hist[0] = data[0], Hist[1] = data[1], Hist[2] = data[2], count = 0; TRUE; Hist[3] = data[3]){

    treat = 63 & PINC;
    data[3] = treat;

    delay(60);

    //if(!(Serial.available()>0)){
      //Serial.print(data[3]);
      //Serial.print(" , ");
      //Serial.println(count);
    //}

    //timer setup
    if(data[4] == 65){
      count++;
    }else{
      count = 0;
    }
    //catch timer
    if(count == 15000){
      data[3] = 300;
      count = 0;
    }
    //end timer

    /*********************/
    if(data[3] == Hist[3])
      continue;
    //if(!(Serial.available()>0)){
      //Serial.println(data[3]);
    //}
//    if(data[1] == data[3])
//      continue;
    /*********************/
    /*********************Search and apply changes*******************/
    for( l = 0, c = 0; l < lines   ; l++){
      if(c >= column){
        l--;
        data[4] = mem[l][c];
        //update
        data[0] = data[2];
        data[1] = data[3];
        data[2] = data[4];
        Hist[0] = data[0];
        Hist[1] = data[1];
        Hist[2] = data[2];
        Hist[4] = data[4];
        //send
        PORTB = 63 & data[4];
        break;
      }
      //startup c=2, c=0, for more precise.
      for(c=2; c < column; c++){
        if(data[c] == mem[l][c]){
          continue;
        }else{
```

```
        break;
      }
    }
  }
  /**************************FALL THREW***************************************/




  /*****************************************************************************/
 }
}
//Nao existe futuro apenas present proximo.
//The more in depth the state machine is the more obidiente and dumb it becomes, because
//it will have to specify all possible cases.
```

```
//Este programa é aplicado para botoneiras start/stop,
//aplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemialnove.
#include<avr/io.h>
#include<avr/interrupt.h>
#define TRUE 1
#define FALSE 0
#define column 2
#define lines 52

void setup()
{
  // start serial port at 9600 bps:
  //Serial.begin(9600);
  DDRC = B00000000;
  PORTC = B11111111;
  DDRB = B11111111;
  PORTB = B00000000;
}

void loop()
{
  int i, l, c;
  int data[3];
  int Hist[3];
  int treat;
  int count=0;
  //inic key
  data[0] = 0;
  data[2] = 0;
  //mem prepared for depth 2 in FSM (finite state machine).
  const int mem[(lines+1)][(column+1)]=
  {
    {65,300, 0},//timer off output 0
    { 0,21, 0},//1
    {65,21,65},//2
    { 0,17,65},//3timer on and output 1
    {65,17,65},//4
    { 0,20,65},//5
    {65,20,65},//6
    { 0, 5,65},//7
    {65, 5,65},//8
    {65,29, 0},//9
    { 0,29, 0},//10
    {65,23, 0},//11
    { 0,23, 0},//12
    {65,31, 0},//13
    { 0,31, 0},//14
    {65, 7, 0},//15
    { 0, 7, 0},//16
    {65,13, 0},//17
    { 0,13, 0},//18
    {65,37, 0},//19
    { 0,37, 0},//20
    {65,28, 0},//21
    { 0,28, 0},//22
    {65,25, 0},//23
    { 0,25, 0},//24
    {65,17,65},//25
    { 0,13, 0},//26
    { 0,53, 0},//27
    {65,53, 0},//28
    { 0,53, 0},//29
    {65,22, 0},//30
    { 0,22, 0},//31
    {65,61, 0},//32
    { 0,61, 0},//33
    {65,55, 0},//34
    { 0,55, 0},//35
    {65,63, 0},//36
    { 0,63, 0},//37
    {65,52, 0},//38
    { 0,52, 0},//39
    {65,19, 0},//40
    { 0,19, 0},//41
    {65, 9, 0},//42
    { 0, 9, 0},//43
    {65,12, 0},//44
    { 0,12, 0},//45
    { 0,24, 0},//46
    { 0,24, 0},//47
    {65,49, 0},//48
    { 0,49, 0}//49
  };

  /*******************CICLOS DE MAQUINA*********************/
  for( Hist[0] = data[0], Hist[2] = data[2], count = 0; TRUE; Hist[1] = data[1]){

    treat = 63 & PINC;
    data[1] = treat;

    delay(60);

    //if(!(Serial.available()>0)){
      //Serial.print(data[3]);
      //Serial.print(" , ");
      //Serial.println(count);
    //}

    //timer setup
    if(data[2] == 65){
      count++;
    }else{
      count = 0;
    }
    //catch timer
    if(count == 15000){
      data[1] = 300;
      count = 0;
    }
    //end timer

    /**********************/
    if(data[1] == Hist[1])
      continue;
    //if(!(Serial.available()>0)){
      //Serial.println(data[1]);
    //}
    /*********************/
    /*********************Search and apply changes********************/
    for( l = 0, c = 0; l < lines   ; l++){
      if(c >= column){
        l--;
        data[2] = mem[l][c];
        //update
        data[0] = data[2];
        Hist[0] = data[0];
        //send
        PORTB = 63 & data[2];
        break;
      }
      /***/
      for(c=0; c < column; c++){
        if(data[c] == mem[l][c]){
          continue;
        }else{
          break;
        }
      }
    }
    /*************************FALL THREW*************************************/
```

```
    /******************************************************************************/
  }
}
//The least error prone.
```

```
#include<avr/io.h>
#define TRUE 1
#define FALSE 0
#define BufSize 127
#define C 0
#define B 1

int getnum(char* x);

void setup()
{
  // start serial port at 9600 bps:
  Serial.begin(9600);
  DDRC=B00000000;
  PORTC=B11111111;
  DDRB=B11111111;
  PORTB=B00000000;
}

void loop()
{
  int i;
  int Entry[2];
  int Past[2];
  int Hist[2];
  char IncomingByte;
  char State[BufSize];


  Serial.flush();
  //INIC

  for( Past[C] = Entry[C], Past[B] = Entry[B]; TRUE; Hist[C] = Entry[C], Hist[B] = Entry[B]){
    //ENTRADA Portas
    Entry[C]=PINC;
    //ENTRADA Serial
    if(Serial.available() > 0){
      delay(25);//wait for incoming data.
      for(i = 0; IncomingByte = Serial.read(); i++){
            if((IncomingByte == '\r') || (IncomingByte == '\n')){
              State[i] = '\0';
              Serial.flush();
              break;
            }else{
              State[i]=IncomingByte;
            }
      }
      Entry[B] = getnum(&State[0]);
    }

    if((Entry[C] == Hist[C]) && (Entry[B] == Hist[B]))
      continue;
    for( ; TRUE; Past[C] = Entry[C], Past[B] = Entry[B]){
      if((Entry[C] == Past[C]) && (Entry[B] == Past[B]))
          break;

      /**Processing***/
      if(Entry[C] != Past[C]){
          if(!(Serial.available() > 0)){
            //leituras do microcontrolador.
            Serial.println(Entry[C],DEC);
            delay(10);
          }
      }

      if(Entry[B] != Past[B]){
            PORTB = Entry[B];
      }
    }
  }
}
//have to press reset in learning mode always.
/***FUNCTIONS***/
int getnum(char* x)
{
  int num;
  if(sscanf(x,"%d",&num) != 0){
    if (num == NULL)
      num = 0;
    return num;
  }else{
    return 0;
  }
}
//char *X, X is a variable that stores a fisical
//address with a cast type char.
//getnum works with string terminating with only NULL or '\0'
//my style
//Lesson always be flexible and except and obey the good practices,
// without any desobidience, and for extra culture if desired try to
// clarify the why things are as they are. Never be stuborn.
```

```
#include<avr/io.h>
#define TRUE 1
#define FALSE 0
#define BufSize 127
#define C 0
#define B 1

int getnum(char* x);

void setup()
{
  // start serial port at 9600 bps:
  Serial.begin(9600);
  DDRC=B00000000;
  PORTC=B11111111;
  DDRB=B11111111;
  PORTB=B00000000;
}

void loop()
{
  int i;
  int Entry[2];
  int Past[2];
  int Hist[2];
  char IncomingByte;
  char State[BufSize];


  Serial.flush();
  //INIC

  for( Past[C] = Entry[C], Past[B] = Entry[B]; TRUE; Hist[C] = Entry[C], Hist[B] = Entry[B]){
    //ENTRADA Portas
    Entry[C]=PINC;
    //ENTRADA Serial
    if(Serial.available() > 0){
      delay(25);//wait for incoming data.
      for(i = 0; IncomingByte = Serial.read(); i++){
            if((IncomingByte == '\r') || (IncomingByte == '\n')){
              State[i] = '\0';
              Serial.flush();
              break;
            }else{
              State[i]=IncomingByte;
            }
      }
      Entry[B] = getnum(&State[0]);
    }

    if((Entry[C] == Hist[C]) && (Entry[B] == Hist[B]))
      continue;
    for( ; TRUE; Past[C] = Entry[C], Past[B] = Entry[B]){
      if((Entry[C] == Past[C]) && (Entry[B] == Past[B]))
         break;

      /**Processing***/
      if(Entry[C] != Past[C]){
         if(!(Serial.available() > 0)){
           //leituras do microcontrolador.
           Serial.println(Entry[C],DEC);
           delay(10);
         }
      }

      if(Entry[B] != Past[B]){
           PORTB = Entry[B];
      }
    }
  }
}
//have to press reset in learning mode always.
/***FUNCTIONS***/
int getnum(char* x)
{
  int num;
  if(sscanf(x,"%d",&num) != 0){
    if (num == NULL)
      num = 0;
    return num;
  }else{
    return 0;
  }
}
//char *X, X is a variable that stores a fisical
//address with a cast type char.
//getnum works with string terminating with only NULL or '\0'
//my style
//Lesson always be flexible and except and obey the good practices,
// without any desobidience, and for extra culture if desired try to
// clarify the why things are as they are. Never be stuborn.
```

```
//Este programa é aplicado para botoneiras start/stop,
//eplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemialnove.
#include<avr/io.h>
#define TRUE 1
#define FALSE 0
#define column 4
#define lines 52

void setup()
{
  // start serial port at 9600 bps:
  //Serial.begin(9600);
  DDRC = B00000000;
  PORTC = B11111111;
  DDRB = B11111111;
  PORTB = B00000000;
}

void loop()
{
  int i, l, c;
  int data[5];
  int Hist[5];
  int count;
  //inic key
  data[0] = 0;
  data[1] = 63;
  data[2] = 0;
  data[4] = 0;
  //mem prepared for depth 2 in FSM (finite state machine).
  const int mem[(lines+1)][(column+1)]=
  {
    { 0, 0, 1,300, 0},//timer 23
    { 0, 0, 0,21, 0},//1
    { 0, 0, 1,21, 1},//2
    { 0, 0, 0,17, 1},//3
    { 0, 0, 1,17, 1},//4
    { 0, 0, 0,20, 1},//5
    { 0, 0, 1,20, 1},//6
    { 0, 0, 0, 5, 1},//7
    { 0, 0, 1, 5, 1},//8
    { 0, 0, 1,29, 0},//9
    { 0, 0, 0,29, 0},//10
    { 0, 0, 1,23, 0},//11
    { 0, 0, 0,23, 0},//12
    { 0, 0, 1,31, 0},//13
    { 0, 0, 0,31, 0},//14
    { 0, 0, 1, 7, 0},//15
    { 0, 0, 0, 7, 0},//16
    { 0, 0, 1,13, 0},//17
    { 0, 0, 0,13, 0},//18
    { 0, 0, 1,37, 0},//19
    { 0, 0, 0,37, 0},//20
    { 0, 0, 1,28, 0},//21
    { 0, 0, 0,28, 0},//22
    { 0, 0, 1,25, 0},//23
    { 0, 0, 0,25, 0},//24
    { 0, 0, 1,17, 1},//25
    { 0, 0, 0,13, 0},//26
    { 0, 0, 0,53, 0},//27
    { 0, 0, 1,53, 0},//28
    { 0, 0, 0,53, 0},//29
    { 0, 0, 1,22, 0},//30
    { 0, 0, 0,22, 0},//31
    { 0, 0, 1,61, 0},//32
    { 0, 0, 0,61, 0},//33
    { 0, 0, 1,55, 0},//34
    { 0, 0, 0,55, 0},//35
    { 0, 0, 1,63, 0},//36
    { 0, 0, 0,63, 0},//37
    { 0, 0, 1,52, 0},//38
    { 0, 0, 0,52, 0},//39
    { 0, 0, 1,19, 0},//40
    { 0, 0, 0,19, 0},//41
    { 0, 0, 1, 9, 0},//42
    { 0, 0, 0, 9, 0},//43
    { 0, 0, 1,12, 0},//44
    { 0, 0, 0,12, 0},//45
    { 0, 0, 0,24, 0},//46
    { 0, 0, 0,24, 0},//47
    { 0, 0, 1,49, 0},//48
    { 0, 0, 0,49, 0}//49
  };

  /*******************CICLOS DE MAQUINA*********************/
  for( Hist[0] = data[0], Hist[1] = data[1], Hist[2] = data[2], count = 0; TRUE; Hist[3] = data[3]){

    data[3] = PINC;

    delay(60);

    //if(!(Serial.available()>0)){
      //Serial.print(data[3]);
      //Serial.print(" , ");
      //Serial.println(count);
    //}

    //timer setup
    if(data[4] == 1){
      count++;
    }else{
      count = 0;
    }
    //catch timer
    if(count == 15000){
      data[3] = 300;
      count = 0;
    }
    //end timer

    /**********************/
    if(data[3] == Hist[3])
      continue;
    if(data[1] == data[3])
      continue;
    /*********************/
    /*********************Search and apply changes********************/
    for( l = 0, c = 0; l < lines   ; l++){
      if(c >= column){
        l--;
        data[4] = mem[l][c];
        PORTB = data[4];
        //update
        data[0] = data[2];
        data[1] = data[3];
        data[2] = data[4];
        Hist[0] = data[0];
        Hist[1] = data[1];
        Hist[2] = data[2];
        Hist[4] = data[4];
        break;
      }
      //startup c=2, c=0, for more precise.
      for(c=2; c < column; c++){
        if(data[c] == mem[l][c]){
          continue;
        }else{
          break;
        }
      }
    }
    /************************FALL THREW************************************/
```

```
    /************************************************************************/
  }
}
//Nao existe futuro apenas present proximo.
//The more in depth the state machine is the more obidiente and dumb it becomes, because
//it will have to specify all possible cases.
```

```
//Este programa é aplicado para botoneiras start/stop,
//aplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemialnove.
#include<avr/io.h>
#include<avr/interrupt.h>
#define TRUE 1
#define FALSE 0
#define column 4
#define lines 52

void setup()
{
  // start serial port at 9600 bps:
  //Serial.begin(9600);
  DDRC = B00000000;
  PORTC = B11111111;
  DDRB = B11111111;
  PORTB = B00000000;
}

void loop()
{
  int i, l, c;
  int data[5];
  int Hist[5];
  int count=0;
  //inic key
  data[0] = 0;
  data[1] = 63;
  data[2] = 0;
  data[4] = 0;
  //mem prepared for depth 2 in FSM (finite state machine).
  const int mem[(lines+1)][(column+1)]=
  {
    { 0, 0,65,300, 0},//timer off
    { 0, 0, 0,21, 0},//1
    { 0, 0,65,21,65},//2
    { 0, 0, 0,17,65},//3timer on and output 1
    { 0, 0,65,17,65},//4
    { 0, 0, 0,20,65},//5
    { 0, 0,65,20,65},//6
    { 0, 0, 0, 5,65},//7
    { 0, 0,65, 5,65},//8
    { 0, 0,65,29, 0},//9
    { 0, 0, 0,29, 0},//10
    { 0, 0,65,23, 0},//11
    { 0, 0, 0,23, 0},//12
    { 0, 0,65,31, 0},//13
    { 0, 0, 0,31, 0},//14
    { 0, 0,65, 7, 0},//15
    { 0, 0, 0, 7, 0},//16
    { 0, 0,65,13, 0},//17
    { 0, 0, 0,13, 0},//18
    { 0, 0,65,37, 0},//19
    { 0, 0, 0,37, 0},//20
    { 0, 0,65,28, 0},//21
    { 0, 0, 0,28, 0},//22
    { 0, 0,65,25, 0},//23
    { 0, 0, 0,25, 0},//24
    { 0, 0,65,17,65},//25
    { 0, 0, 0,13, 0},//26
    { 0, 0, 0,53, 0},//27
    { 0, 0,65,53, 0},//28
    { 0, 0, 0,53, 0},//29
    { 0, 0,65,22, 0},//30
    { 0, 0, 0,22, 0},//31
    { 0, 0,65,61, 0},//32
    { 0, 0, 0,61, 0},//33
    { 0, 0,65,55, 0},//34
    { 0, 0, 0,55, 0},//35
    { 0, 0,65,63, 0},//36
    { 0, 0, 0,63, 0},//37
    { 0, 0,65,52, 0},//38
    { 0, 0, 0,52, 0},//39
    { 0, 0,65,19, 0},//40
    { 0, 0, 0,19, 0},//41
    { 0, 0,65, 9, 0},//42
    { 0, 0, 0, 9, 0},//43
    { 0, 0,65,12, 0},//44
    { 0, 0, 0,12, 0},//45
    { 0, 0, 0,24, 0},//46
    { 0, 0, 0,24, 0},//47
    { 0, 0,65,49, 0},//48
    { 0, 0, 0,49, 0}//49
  };

  /*******************CICLOS DE MAQUINA*********************/
  for( Hist[0] = data[0], Hist[1] = data[1], Hist[2] = data[2], count = 0; TRUE; Hist[3] = data[3]){

    data[3] = 63 & PINC;

    delay(60);

    //if(!(Serial.available()>0)){
      //Serial.print(data[3]);
      //Serial.print(" , ");
      //Serial.println(count);
    //}

    //timer setup
    if(data[4] == 65){
      count++;
    }else{
      count = 0;
    }
    //catch timer
    if(count == 15000){
      data[3] = 300;
      count = 0;
    }
    //end timer

    /**********************/
    if(data[3] == Hist[3])
      continue;
    if(data[1] == data[3])
      continue;
    /*********************/
    /*********************Search and apply changes********************/
    for( l = 0, c = 0; l < lines   ; l++){
      if(c >= column){
        l--;
        data[4] = mem[l][c];
        //update
        data[0] = data[2];
        data[1] = data[3];
        data[2] = data[4];
        Hist[0] = data[0];
        Hist[1] = data[1];
        Hist[2] = data[2];
        Hist[4] = data[4];
        //send
        PORTB = 63 & data[4];
        break;
      }
      //startup c=2, c=0, for more precise.
      for(c=2; c < column; c++){
        if(data[c] == mem[l][c]){
          continue;
        }else{
          break;
        }
      }
    }
    /************************FALL THREW*************************************/
```

```
    /**************************************************************************/
  }
}
//Nao existe futuro apenas present proximo.
//The more in depth the state machine is the more obidiente and dumb it becomes, because
//it will have to specify all possible cases.
```

```
//Este programa é aplicado para botoneiras start/stop,
//aplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemialnove.
#include<avr/io.h>
#include<avr/interrupt.h>
#define TRUE 1
#define FALSE 0
#define column 2
#define lines 52

void setup()
{
  // start serial port at 9600 bps:
  //Serial.begin(9600);
  DDRC = B00000000;
  PORTC = B11111111;
  DDRB = B11111111;
  PORTB = B00000000;
}

void loop()
{
  int i, l, c;
  int data[3];
  int Hist[3];
  int treat;
  int count=0;
  //inic key
  data[0] = 0;
  data[2] = 0;
  //mem prepared for depth 2 in FSM (finite state machine).
  const int mem[(lines+1)][(column+1)]=
  {
    {65,300, 0},//timer off output 0
    { 0,21, 0},//1
    {65,21,65},//2
    { 0,17,65},//3timer on and output 1
    {65,17,65},//4
    { 0,20,65},//5
    {65,20,65},//6
    { 0, 5,65},//7
    {65, 5,65},//8
    {65,29, 0},//9
    { 0,29, 0},//10
    {65,23, 0},//11
    { 0,23, 0},//12
    {65,31, 0},//13
    { 0,31, 0},//14
    {65, 7, 0},//15
    { 0, 7, 0},//16
    {65,13, 0},//17
    { 0,13, 0},//18
    {65,37, 0},//19
    { 0,37, 0},//20
    {65,28, 0},//21
    { 0,28, 0},//22
    {65,25, 0},//23
    { 0,25, 0},//24
    {65,17,65},//25
    { 0,13, 0},//26
    { 0,53, 0},//27
    {65,53, 0},//28
    { 0,53, 0},//29
    {65,22, 0},//30
    { 0,22, 0},//31
    {65,61, 0},//32
    { 0,61, 0},//33
    {65,55, 0},//34
    { 0,55, 0},//35
    {65,63, 0},//36
    { 0,63, 0},//37
    {65,52, 0},//38
    { 0,52, 0},//39
    {65,19, 0},//40
    { 0,19, 0},//41
    {65, 9, 0},//42
    { 0, 9, 0},//43
    {65,12, 0},//44
    { 0,12, 0},//45
    { 0,24, 0},//46
    { 0,24, 0},//47
    {65,49, 0},//48
    { 0,49, 0}//49
  };

  /*******************CICLOS DE MAQUINA*********************/
  for( Hist[0] = data[0], Hist[2] = data[2], count = 0; TRUE; Hist[1] = data[1]){

    treat = 63 & PINC;
    data[1] = treat;

    delay(60);

    //if(!(Serial.available()>0)){
      //Serial.print(data[3]);
      //Serial.print(" , ");
      //Serial.println(count);
    //}

    //timer setup
    if(data[2] == 65){
      count++;
    }else{
      count = 0;
    }
    //catch timer
    if(count == 15000){
      data[1] = 300;
      count = 0;
    }
    //end timer

    /*********************/
    if(data[1] == Hist[1])
      continue;
    //if(!(Serial.available()>0)){
      //Serial.println(data[1]);
    //}
    /********************/
    /*********************Search and apply changes********************/
    for( l = 0, c = 0; l < lines   ; l++){
      if(c >= column){
        l--;
        data[2] = mem[l][c];
        //update
        data[0] = data[2];
        Hist[0] = data[0];
        //send
        PORTB = 63 & data[2];
        break;
      }
      /***/
      for(c=0; c < column; c++){
        if(data[c] == mem[l][c]){
          continue;
        }else{
          break;
        }
      }
    }
    /************************FALL THREW*************************************/
```

```
    /************************************************************************/
  }
}
//The least error prone.
```

```
//Este programa é aplicado para botoneiras start/stop,
//aplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemialnove.
#include<avr/io.h>
#include<avr/interrupt.h>
#define TRUE 1
#define FALSE 0
#define column 2
#define lines 52

void setup()
{
  // start serial port at 9600 bps:
  //Serial.begin(9600);
  DDRC = B00000000;
  PORTC = B11111111;
  DDRB = B11111111;
  PORTB = B00000000;
}

void loop()
{
  int i, l, c;
  int data[3];
  int Hist[2];
  int treat;
  int count=0;
  //inic key
  data[0] = 0;
  data[2] = 0;
  //mem prepared for depth 2 in FSM (finite state machine).
  const int mem[(lines+1)][(column+1)]=
  {
    {65,300, 0},//timer off output 0
    { 0,21, 0},//1
    {65,21,65},//2
    { 0,17,65},//3timer on and output 1
    {65,17,65},//4
    { 0,20,65},//5
    {65,20,65},//6
    { 0, 5,65},//7
    {65, 5,65},//8
    {65,29, 0},//9
    { 0,29, 0},//10
    {65,23, 0},//11
    { 0,23, 0},//12
    {65,31, 0},//13
    { 0,31, 0},//14
    {65, 7, 0},//15
    { 0, 7, 0},//16
    {65,13, 0},//17
    { 0,13, 0},//18
    {65,37, 0},//19
    { 0,37, 0},//20
    {65,28, 0},//21
    { 0,28, 0},//22
    {65,25, 0},//23
    { 0,25, 0},//24
    {65,17,65},//25
    { 0,13, 0},//26
    { 0,53, 0},//27
    {65,53, 0},//28
    { 0,53, 0},//29
    {65,22, 0},//30
    { 0,22, 0},//31
    {65,61, 0},//32
    { 0,61, 0},//33
    {65,55, 0},//34
    { 0,55, 0},//35
    {65,63, 0},//36
    { 0,63, 0},//37
    {65,52, 0},//38
    { 0,52, 0},//39
    {65,19, 0},//40
    { 0,19, 0},//41
    {65, 9, 0},//42
    { 0, 9, 0},//43
    {65,12, 0},//44
    { 0,12, 0},//45
    { 0,24, 0},//46
    { 0,24, 0},//47
    {65,49, 0},//48
    { 0,49, 0}//49
  };

  /*******************CICLOS DE MAQUINA*********************/
  for( Hist[0] = data[0], count = 0; TRUE; Hist[1] = data[1]){

    treat = 63 & PINC;
    data[1] = treat;

    delay(60);

    //if(!(Serial.available()>0)){
      //Serial.print(data[3]);
      //Serial.print(" , ");
      //Serial.println(count);
    //}

    //timer setup
    if(data[2] == 65){
      count++;
    }else{
      count = 0;
    }
    //catch timer
    if(count == 15000){
      data[1] = 300;
      count = 0;
    }
    //end timer

    /**********************/
    if(data[1] == Hist[1])
      continue;
    //if(!(Serial.available()>0)){
      //Serial.println(data[1]);
    //}
    /*********************/
    /*********************Search and apply changes********************/
    for( l = 0, c = 0; l < lines   ; l++){
      if(c >= column){
        l--;
        data[2] = mem[l][c];
        //update
        data[0] = data[2];
        if(data[1] > 63)
          data[1] = Hist[1];
        break;
      }
      /***/
      for(c=0; c < column; c++){
        if(data[c] == mem[l][c]){
          continue;
        }else{
          break;
        }
      }
    }
    /************************FALL THREW**************************************/
    PORTB = 63 & data[2];
```

```
    /***************************************************************************/
  }
}
//The least error prone.
```

```
//Este programa é aplicado para botoneiras start/stop,
//aplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemialnove.
#include<avr/io.h>
#include<avr/interrupt.h>
#define TRUE 1
#define FALSE 0
#define column 2
#define lines 52

void setup()
{
  // start serial port at 9600 bps:
  Serial.begin(9600);
  DDRC = B00000000;
  PORTC = B11111111;
  DDRB = B11111111;
  PORTB = B00000000;
}

void loop()
{
  int i, l, c;
  int data[3];
  int Hist[2];
  int treat;
  int count_1 = 0;
  int comp_1 = 10;
  int count_2 = 0;
  int comp_2 = 10;
  //inic key
  data[0] = 0;
  data[2] = 0;
  //mem prepared for depth 2 in FSM (finite state machine).
  const int mem[(lines+1)][(column+1)]=
  {
    {0,62,65},//timer off output 0
    {65,300,64},//1
    {64,400,65},//2
    {65,62,0},//3timer on and output 1
    {64,62,0},//4
    {65,61,129},//5
    {65,59,257},//6
    {64,61,128},//7
    {64,59.256}//8
  };
```

```
/*******************CICLOS DE MAQUINA*********************/
 for( Hist[0] = data[0], count_1 = 0, count_2 = 0; TRUE; Hist[1] = data[1]){

  treat = 63 & PINC;
  data[1] = treat;

  delay(60);

  //timer_1 setup
  if(data[2] == 65){
   count_1++;
  }else{
   count_1 = 0;
  }
  //catch timer
  if(count_1 == comp_1){
   data[1] = 300;
   count_1 = 0;
  }
  //end timer

  //timer_2 setup
  if(data[2] == 64){
   count_2++;
  }else{
   count_2 = 0;
  }
  //catch timer
  if(count_2 == comp_2){
   data[1] = 400;
   count_2 = 0;
  }
  //end timer

  if(data[2] == 128 || data[2] == 129){
   if(count_1 < (2*comp_1)){
    comp_1++;
    comp_2--;
   }
  }

  if(data[2] == 256 || data[2] == 257){
   if(count_2 < (2*comp_1)){
    comp_2++;
    comp_1--;
   }
  }
```

```
    if(!(Serial.available()>0)){
      Serial.print(data[1]);
      Serial.print(" , ");
      Serial.print(count_1);
      Serial.print(" , ");
      Serial.println(count_2);
    }

    /*********************/
    if(data[1] == Hist[1])
      continue;

    /*********************/
/*********************Search and apply changes********************/
    for( l = 0, c = 0; l < lines   ; l++){
      if(c >= column){
        l--;
        data[2] = mem[l][c];
        //update
        data[0] = data[2];
        if(data[1] > 63)
          data[1] = Hist[1];
        break;
      }
      /***/
      for(c=0; c < column; c++){
        if(data[c] == mem[l][c]){
          continue;
        }else{
          break;
        }
      }
    }
/************************FALL THREW**************************************/
    PORTB = 63 & data[2];





/*************************************************************************/
  }
}
//The least error prone.
```

```
//Este programa é aplicado para botoneiras start/stop,
//aplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemialnove.
#include<avr/io.h>
#include<avr/interrupt.h>
#define TRUE 1
#define FALSE 0
#define column 2
#define lines 52

void setup()
{
  // start serial port at 9600 bps:
  Serial.begin(9600);
  DDRC = B00000000;
  PORTC = B11111111;
  DDRB = B11111111;
  PORTB = B00000000;
}

void loop()
{
  int i, l, c;
  int data[3];
  int Hist[2];
  int treat;
  int count_1 = 0;
  int comp_1 = 10;
  int count_2 = 0;
  int comp_2 = 10;
  //inic key
  data[0] = 0;
  data[2] = 0;
  //mem prepared for depth 2 in FSM (finite state machine).
  const int mem[(lines+1)][(column+1)]=
  {
    {0,62,65},//timer off output 0
    {65,300,64},//1
    {64,400,65},//2
    {65,62,0},//3timer on and output 1
    {64,62,0}//4

  };

/*******************CICLOS DE MAQUINA*********************/
  for( Hist[0] = data[0], count_1 = 0, count_2 = 0; TRUE; Hist[1] = data[1]){
```

```
treat = 63 & PINC;
data[1] = treat;

delay(60);


if((data[1] == 61) && (data[1] != Hist[1])){
  if((count_1 < 20) && (count_1 < comp_1)){
    comp_1++;
    comp_2--;
  }
  continue;
}

if((data[1] == 59) && (data[1] != Hist[1])){
  if((count_2 < 20) && (count_2 < comp_2)){
    comp_2++;
    comp_1--;
  }
  continue;
}


//timer_1 setup
if(count_1 == comp_1){
  data[1] = 300;
  count_1 = 0;
}
if(data[2] == 65){
  count_1++;
}else{
  count_1 = 0;
}
//end timer

//timer_2 setup
if(count_2 == comp_2){
  data[1] = 400;
  count_2 = 0;
}
if(data[2] == 64){
  count_2++;
}else{
  count_2 = 0;
}
//end timer

   if(!(Serial.available()>0)){
  Serial.print(data[1]);
```

```
      Serial.print(" , ");
      Serial.print(count_1);
      Serial.print(" , ");
      Serial.println(count_2);
    }

    /**********************/
    if(data[1] == Hist[1])
      continue;

    /*********************/
/*********************Search and apply changes********************/
    for( l = 0, c = 0; l < lines   ; l++){
      if(c >= column){
        l--;
        data[2] = mem[l][c];
        //update
        data[0] = data[2];
        if(data[1] > 63)
          data[1] = Hist[1];
        break;
      }
      /***/
      for(c=0; c < column; c++){
        if(data[c] == mem[l][c]){
          continue;
        }else{
          break;
        }
      }
    }
/************************FALL THREW*************************************/
    PORTB = 63 & data[2];





/*************************************************************************/
  }
}
//The least error prone.
```

```
//Este programa é aplicado para botoneiras start/stop,
//aplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemialnove.
#include<avr/io.h>
#include<avr/interrupt.h>
#define TRUE 1
#define FALSE 0
#define column 2
#define lines 52

void setup()
{
  // start serial port at 9600 bps:
  Serial.begin(9600);
  DDRC = B00000000;
  PORTC = B11111111;
  DDRB = B11111111;
  PORTB = B00000000;
}

void loop()
{
  int i, l, c;
  int data[3];
  int Hist[2];
  int treat;
  int count_1 = 0;
  int comp_1 = 10;
  int count_2 = 0;
  int comp_2 = 10;
  //inic key
  data[0] = 0;
  data[2] = 0;
  //mem prepared for depth 2 in FSM (finite state machine).
  const int mem[(lines+1)][(column+1)]=
  {
    {0,62,65},//timer off output 0
    {65,300,64},//1
    {64,400,65},//2
    {65,62,0},//3timer on and output 1
    {64,62,0}//4

  };

/******************CICLOS DE MAQUINA********************/
  for( Hist[0] = data[0], count_1 = 0, count_2 = 0; TRUE; Hist[1] = data[1]){
    treat = 63 & PINC;
    data[1] = treat;

    delay(300);

    if(data[2] == 65){
      count_1++;
    }
    if(count_1==comp_1){
      data[1]=300;
      count_1=0;
    }

    if(data[2] == 64){
      count_2++;
    }
    if(count_2==comp_2){
      data[1]=400;
      count_2=0;
    }



    if(!(Serial.available()>0)){
      Serial.print(data[1]);
      Serial.print(" , ");
      Serial.print(count_1);
      Serial.print(" , ");
      Serial.println(count_2);
    }

    /*********************/
    if(data[1] == Hist[1])
      continue;

    /********************/
/********************Search and apply changes********************/
    for( l = 0, c = 0; l < lines   ; l++){
      if(c >= column){
        l--;
        data[2] = mem[l][c];
        //update
        data[0] = data[2];
        if(data[1] > 63){
          data[1]=Hist[1];
        }
        break;
      }
      /***/
```

```
   for(c=0; c < column; c++){
    if(data[c] == mem[l][c]){
     continue;
    }else{
     break;
    }
   }
  }
/*************************FALL THREW*************************************/
  PORTB = 63 & data[2];
  /******************/
  if(data[1] == 61){
   if((comp_2 > 0)){
    comp_1++;
    comp_2--;
   }
  }
  if(data[1] == 59){
   if((comp_1 > 0)){
    comp_1--;
    comp_2++;
   }
  }





/***********************************************************************/
 }
}
//The least error prone.
```

```
//Este programa é aplicado para botoneiras start/stop,
//aplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemialnove.
#include<avr/io.h>
#include<avr/interrupt.h>
#define TRUE 1
#define FALSE 0
#define column 2
#define lines 52

void setup()
{
 // start serial port at 9600 bps:
 Serial.begin(9600);
 DDRC = B00000000;
 PORTC = B11111111;
 DDRB = B11111111;
 PORTB = B00000000;
}

void loop()
{
 int i, l, c;
 int data[3];
 int Hist[2];
 int treat;
 int count_1 = 0;
 int timon=0;
 int timoff=19;
 //inic key
 data[0] = 0;
 data[2] = 0;
 //mem prepared for depth 2 in FSM (finite state machine).
 const int mem[(lines+1)][(column+1)]=
 {
  {0,62,65},//timer off output 0
  {65,300,64},//1
  {64,400,65},//2
  {65,62,0},//3timer on and output 1
  {64,62,0}//4

 };

/*******************CICLOS DE MAQUINA*********************/
 for( Hist[0] = data[0], count_1 = 0; TRUE; Hist[1] = data[1]){

  treat = 63 & PINC;
```

```
    data[1] = treat;

    delay(60);

    count_1++;
    if(count_1 == 20)
      count_1=0;


    if(count_1 == timoff)
      data[1]=300;
    if(count_1 == timon)
      data[1]=400;


    if(data[1] == 61){
      timon++;
      timoff--;
    }
    if(data[1] == 59){
      timoff++;
      timon--;
    }

    if(!(Serial.available()>0)){
      Serial.print(data[1]);
      Serial.print(" , ");
      Serial.print(count_1);
      Serial.print(" , ");
      Serial.print(timon);
      Serial.print(" , ");
      Serial.println(timoff);
    }

    /*********************/
    if(data[1] == Hist[1])
      continue;

    /*********************/
/*********************Search and apply changes********************/
    for( l = 0, c = 0; l < lines   ; l++){
      if(c >= column){
        l--;
        data[2] = mem[l][c];
        //update
        data[0] = data[2];
        if(data[1] > 63){
          data[1] = Hist[1];
        }
        break;
      }
      /***/
      for(c=0; c < column; c++){
        if(data[c] == mem[l][c]){
          continue;
        }else{
          break;
        }
      }
    }
/*************************FALL THREW**************************************/
    PORTB = 63 & data[2];
    /******************/




/************************************************************************/
  }
}
//The least error prone.
```

```
//Este programa é aplicado para botoneiras start/stop,
//aplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemialnove.
#include<avr/io.h>
#include<avr/interrupt.h>
#define TRUE 1
#define FALSE 0
#define column 2
#define lines 52

void setup()
{
  // start serial port at 9600 bps:
  Serial.begin(9600);
  DDRC = B00000000;
  PORTC = B11111111;
  DDRB = B11111111;
  PORTB = B00000000;
}

void loop()
{
  int i, l, c;
  int data[3];
  int Hist[2];
  int treat;
  int count_1 = 0;
  int timon=0;
  int timoff=19;
  //inic key
  data[0] = 0;
  data[2] = 0;
  //mem prepared for depth 2 in FSM (finite state machine).
  const int mem[(lines+1)][(column+1)]=
  {
    {0,62,65},//timer off output 0
    {65,300,64},//1
    {64,400,65},//2
    {65,62,0},//3timer on and output 1
    {64,62,0}//4

  };

/*******************CICLOS DE MAQUINA*********************/
  for( Hist[0] = data[0], count_1 = 0; TRUE; Hist[1] = data[1]){

    treat = 63 & PINC;
    data[1] = treat;

    delay(60);

    count_1++;
    if(count_1 == 20)
      count_1=0;


    if(count_1 == timoff)
      data[1]=300;
    if(count_1 == timon)
      data[1]=400;


    if(!(Serial.available()>0)){
      Serial.print(data[1]);
      Serial.print(" , ");
      Serial.print(count_1);
      Serial.print(" , ");
      Serial.print(timon);
      Serial.print(" , ");
      Serial.println(timoff);
    }

    /*********************/
    if(data[1] == Hist[1])
      continue;

    /********************/
/*********************Search and apply changes********************/
    for( l = 0, c = 0; l < lines   ; l++){
      if(c >= column){
        l--;
        data[2] = mem[l][c];
        //update
        data[0] = data[2];
        break;
      }
      /***/
      for(c=0; c < column; c++){
        if(data[c] == mem[l][c]){
          continue;
        }else{
          break;
        }
      }
    }
/************************FALL THREW*************************************/
```

```
    PORTB = 63 & data[2];
    /******************/
    if(data[1] > 63){
          data[1] = Hist[1];
          continue;
    }
    /******************/
    if(data[1] == 61){
      timon++;
      timoff--;
    }
    if(data[1] == 59){
      timoff++;
      timon--;
    }




/**********************************************************************/
  }
}
//The least error prone.
```

```
//Este programa é aplicado para botoneiras start/stop,
//aplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemialnove.
#include<avr/io.h>
#include<avr/interrupt.h>
#define TRUE 1
#define FALSE 0
#define column 2
#define lines 52

void setup()
{
  // start serial port at 9600 bps:
  //Serial.begin(9600);
  DDRC = B00000000;
  PORTC = B11111111;
  DDRB = B11111111;
  PORTB = B00000000;
}

void loop()
{
  int i, l, c;
  int data[3];
  int Hist[2];
  int treat;
  int count_1 = 0;
  int timon=0;
  int timoff=19;
  //inic key
  data[0] = 0;
  data[2] = 0;
  //mem prepared for depth 2 in FSM (finite state machine).
  const int mem[(lines+1)][(column+1)]=
  {
    {0,62,65},//timer off output 0
    {65,300,64},//1
    {64,400,65},//2
    {65,62,0},//3timer on and output 1
    {64,62,0}//4

  };

/*******************CICLOS DE MAQUINA*********************/
  for( Hist[0] = data[0], count_1 = 0; TRUE; Hist[1] = data[1]){

    treat = 63 & PINC;
    data[1] = treat;

    delay(10);

    count_1++;
    if(count_1 == 20)
      count_1=0;


    if(count_1 == timoff)
      data[1]=300;
    if(count_1 == timon)
      data[1]=400;


    //if(!(Serial.available()>0)){
      //Serial.print(data[1]);
      //Serial.print(" , ");
      //Serial.print(count_1);
      //Serial.print(" , ");
      //Serial.print(timon);
      //Serial.print(" , ");
      //Serial.println(timoff);
    //}

    /**********************/
    if(data[1] == Hist[1])
      continue;

    /********************/
/********************Search and apply changes********************/
    for( l = 0, c = 0; l < lines   ; l++){
      if(c >= column){
        l--;
        data[2] = mem[l][c];
        //update
        data[0] = data[2];
        break;
      }
      /***/
      for(c=0; c < column; c++){
        if(data[c] == mem[l][c]){
          continue;
        }else{
          break;
        }
      }
    }
/***********************FALL THREW************************************/
```

```
  PORTB = 63 & data[2];
  /******************/
  if(data[1] > 63){
      data[1] = Hist[1];
      continue;
  }
  /******************/
  if(data[1] == 61){
    timon++;
    timoff--;
  }
  if(data[1] == 59){
    timoff++;
    timon--;
  }




/************************************************************************/
  }
}
//The least error prone.
```

```
//Este programa é aplicado para botoneiras start/stop,
//aplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemialnove.
#include<avr/io.h>
#include<avr/interrupt.h>
#define TRUE 1
#define FALSE 0
#define column 2
#define lines 52

void setup()
{
  // start serial port at 9600 bps:
  //Serial.begin(9600);
  DDRC = B00000000;
  PORTC = B11111111;
  DDRB = B11111111;
  PORTB = B00000000;
}

void loop()
{
  int i, l, c;
  int data[3];
  int Hist[2];
  int treat;
  int count_1 = 0;
  int timon=0;
  int timoff=9;
  //inic key
  data[0] = 0;
  data[2] = 0;
  //mem prepared for depth 2 in FSM (finite state machine).
  const int mem[(lines+1)][(column+1)]=
  {
    {0,62,65},//timer off output 0
    {65,300,64},//1
    {64,400,65},//2
    {65,62,0},//3timer on and output 1
    {64,62,0}//4

  };

/*******************CICLOS DE MAQUINA*********************/
  for( Hist[0] = data[0], count_1 = 0; TRUE; Hist[1] = data[1]){

    treat = 63 & PINC;
    data[1] = treat;

    delay(3);

    count_1++;
    if(count_1 == 10)
      count_1=0;


    if(count_1 == timoff)
      data[1]=300;
    if(count_1 == timon)
      data[1]=400;


    //if(!(Serial.available()>0)){
      //Serial.print(data[1]);
      //Serial.print(" , ");
      //Serial.print(count_1);
      //Serial.print(" , ");
      //Serial.print(timon);
      //Serial.print(" , ");
      //Serial.println(timoff);
    //}

    /**********************/
    if(data[1] == Hist[1])
      continue;

    /*********************/
/*********************Search and apply changes********************/
    for( l = 0, c = 0; l < lines   ; l++){
      if(c >= column){
        l--;
        data[2] = mem[l][c];
        //update
        data[0] = data[2];
        break;
      }
      /***/
      for(c=0; c < column; c++){
        if(data[c] == mem[l][c]){
          continue;
        }else{
          break;
        }
      }
    }
/************************FALL THREW*************************************/
```

```
    PORTB = 63 & data[2];
    /*******************/
    if(data[1] > 63){
          data[1] = Hist[1];
          continue;
    }
    /*******************/
    if(data[1] == 61){
      timon++;
      timoff--;
    }
    if(data[1] == 59){
      timoff++;
      timon--;
    }




/******************************************************************************************/
  }
}
//The least error prone.
```

```
//Este programa é aplicado para botoneiras start/stop,
//aplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemialnove.
#include<avr/io.h>
#include<avr/interrupt.h>
#define TRUE 1
#define FALSE 0
#define column 2
#define lines 52

void setup()
{
  // start serial port at 9600 bps:
  //Serial.begin(9600);
  DDRC = B00000000;
  PORTC = B11111111;
  DDRB = B11111111;
  PORTB = B00000000;
}

void loop()
{
  int i, l, c;
  int data[3];
  int Hist[2];
  int treat;
  int count_1 = 0;
  int timon=0;
  int timoff=255;
  //inic key
  data[0] = 0;
  data[2] = 0;
  //mem prepared for depth 2 in FSM (finite state machine).
  const int mem[(lines+1)][(column+1)]=
  {
    {0,62,65},//timer off output 0
    {65,300,64},//1
    {64,400,65},//2
    {65,62,0},//3timer on and output 1
    {64,62,0}//4

  };

/*******************CICLOS DE MAQUINA*********************/
  for( Hist[0] = data[0], count_1 = 0; TRUE; Hist[1] = data[1]){

    treat = 63 & PINC;
    data[1] = treat;

    delayMicroseconds(60);

    count_1++;
    if(count_1 == 256)
      count_1=0;


    if(count_1 == timoff)
      data[1]=300;
    if(count_1 == timon)
      data[1]=400;


    //if(!(Serial.available()>0)){
      //Serial.print(data[1]);
      //Serial.print(" , ");
      //Serial.print(count_1);
      //Serial.print(" , ");
      //Serial.print(timon);
      //Serial.print(" , ");
      //Serial.println(timoff);
    //}

    /**********************/
    if(data[1] == Hist[1])
      continue;

    /********************/
/*********************Search and apply changes********************/
    for( l = 0, c = 0; (l < lines) && (c < column); l++){
      for(c=0; c < column; c++){
        if(data[c] == mem[l][c]){
          continue;
        }else{
          break;
        }
      }
    }
    if(c == column){
        l--;
        data[2] = mem[l][c];
        //update
        data[0] = data[2];
    }
/************************FALL THREW************************************/
    PORTB = 63 & data[2];
    /*****************/
    if(data[1] > 63){
          data[1] = Hist[1];
          continue;
    }
    /*******************/
    if((data[1] == 61) && (timon < 127)){
      timon++;
      timoff--;
    }
    if((data[1] == 59) && (timon > 0) ){
      timoff++;
      timon--;
    }




/*************************************************************************/
  }
}
//The least error prone.
```

```
//Este programa é aplicado para botoneiras start/stop,
//aplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemialnove.
#include<avr/io.h>
#include<avr/interrupt.h>
#define TRUE 1
#define FALSE 0
#define column 2
#define lines 52

void setup()
{
  // start serial port at 9600 bps:
  //Serial.begin(9600);
  DDRC = B00000000;
  PORTC = B11111111;
  DDRB = B11111111;
  PORTB = B00000000;
}

void loop()
{
  int i, l, c;
  int data[3];
  int Hist[2];
  int treat;
  int count_1 = 0;
  int timon=0;
  int timoff=100;
  //inic key
  data[0] = 0;
  data[2] = 0;
  //mem prepared for depth 2 in FSM (finite state machine).
  const int mem[(lines+1)][(column+1)]=
  {
    {0,62,65},//timer off output 0
    {65,300,64},//1
    {64,400,65},//2
    {65,62,0},//3timer on and output 1
    {64,62,0}//4

  };

/*******************CICLOS DE MAQUINA*********************/
  for( Hist[0] = data[0], count_1 = 0; TRUE; Hist[1] = data[1]){

    treat = 63 & PINC;
    data[1] = treat;

    delayMicroseconds(250);

    count_1++;
    if(count_1 == 101)
      count_1=0;


    if(count_1 == timoff)
      data[1]=300;
    if(count_1 == timon)
      data[1]=400;


    //if(!(Serial.available()>0)){
      //Serial.print(data[1]);
      //Serial.print(" , ");
      //Serial.print(count_1);
      //Serial.print(" , ");
      //Serial.print(timon);
      //Serial.print(" , ");
      //Serial.println(timoff);
    //}

    /**********************/
    if(data[1] == Hist[1])
      continue;

    /********************/
/*********************Search and apply changes********************/
    for( l = 0, c = 0; (l < lines) && (c < column); l++){
      for(c=0; c < column; c++){
        if(data[c] == mem[l][c]){
          continue;
        }else{
          break;
        }
      }
    }
    if(c == column){
        l--;
        data[2] = mem[l][c];
        //update
        data[0] = data[2];
    }
/************************FALL THREW************************************/
    PORTB = 63 & data[2];
    /******************/
    if(data[1] > 63){
          data[1] = Hist[1];
          continue;
    }
    /******************/
    if((data[1] == 61) && (timon < 50)){
      timon++;
      timoff--;
    }
    if((data[1] == 59) && (timon > 0) ){
      timoff++;
      timon--;
    }





/**********************************************************************/
  }
}
//The least error prone.
```

```
#include<avr/io.h>
#define TRUE 1
#define FALSE 0
#define BufSize 127
#define A 0
#define C 1

int getnum(char* x);

void setup()
{
  // start serial port at 9600 bps:
  Serial.begin(9600);
  DDRA=B00000000;
  PORTA=B11111111;
  DDRC=B11111111;
  PORTC=B00000000;
}

void loop()
{
  int i;
  int Entry[2];
  int Past[2];
  int Hist[2];
  char IncomingByte;
  char State[BufSize];


  Serial.flush();
  //INIC

  for( Past[A] = Entry[A], Past[C] = Entry[C]; TRUE; Hist[A] = Entry[A], Hist[C] = Entry[C]){
    //ENTRADA Portas
    Entry[A]=PINA;
    //ENTRADA Serial
    if(Serial.available() > 0){
      delay(25);//wait for incoming data.
      for(i = 0; IncomingByte = Serial.read(); i++){
            if((IncomingByte == '\r') || (IncomingByte == '\n')){
              State[i] = '\0';
              Serial.flush();
              break;
            }else{
              State[i]=IncomingByte;
            }
      }
      Entry[C] = getnum(&State[0]);
    }

    if((Entry[A] == Hist[A]) && (Entry[C] == Hist[C]))
      continue;
    for( ; TRUE; Past[A] = Entry[A], Past[C] = Entry[C]){
      if((Entry[A] == Past[A]) && (Entry[C] == Past[C]))
           break;

      /**Processing***/
      if(Entry[A] != Past[A]){
           if(!(Serial.available() > 0)){
             //leituras do microcontrolador.
             Serial.println(Entry[A],DEC);
             delay(10);
           }
      }

      if(Entry[C] != Past[C]){
             PORTC = Entry[C];
      }
    }
  }
}
//have to press reset in learning mode always.
/***FUNCTIONS***/
int getnum(char* x)
{
  int num;
  if(sscanf(x,"%d",&num) != 0){
    if (num == NULL)
      num = 0;
    return num;
  }else{
    return 0;
  }
}
//char* X, X is a variable that stores a fisical
//address with a cast type char.
//getnum works with string terminating with only NULL or '\0'
//my style
```

```
#include<avr/io.h>
#define TRUE 1
#define FALSE 0
#define column 4
#define lines 36

void setup()
{
  // start serial port at 9600 bps:
  Serial.begin(9600);
  DDRA=B00000000;
  PORTA=B11111111;
  DDRC=B11111111;
  PORTC=B00000000;
}

void loop()
{
  int i, l, c;
  int data[5];
  int Hist[5];
  int count=0;

  //inic key
  data[0]=0;
  data[1]=255;
  data[2]=0;
  data[4]=0;
  PORTB = data[2];
  //mem prepared for depth 2 in FSM (finie state machine).
  int mem[36][5]=
  {
    { 0, 0, 0,255, 0},
    { 0, 0, 0,254, 1},
    { 0, 0, 1,253, 0},
    { 0, 0, 1,254, 2},
    { 0, 0, 2,254, 1},
    {0 , 0, 1,300, 0}//timer
  };

  /*******************CICLOS DE MAQUINA*********************/
  for( Hist[0] = data[0], Hist[1] = data[1], Hist[2] = data[2]; TRUE; Hist[3] = data[3]){

      data[3] = PINA;

    delay(60);
```

```
if(!(Serial.available()>0)){
  Serial.print(data[3]);
  Serial.print(", ");
  Serial.println(count);
}

//timer setup
if(data[4]==1){
  count++;
}else{
  count=0;
}
//catch timer
if(count==500){
  data[3]=300;
}
//end timer

/*****/
if(data[3] == Hist[3])
  continue;
/*****/
/*********************Search and apply changes********************/
for( l = 0, c = 0; l < lines   ; l++){
  if(c >= column){
    l--;
    data[4] = mem[l][c];
    //update
    data[0]=data[2];
    data[1]=data[3];
    data[2]=data[4];
    Hist[0]=data[0];
    Hist[1]=data[1];
    Hist[2]=data[2];
    Hist[4]=data[4];
    //send
    PORTC = data[4];
    break;
  }
  //startup c=2, c=0, for more precise.
  for(c=2; c < column; c++){
    if(data[c] == mem[l][c]){
      continue;
    }else{
      break;
    }
  }
}
```

```
  /*****FALL THREW*************************************************************/







  /***************************************************************************/
 }
}
//Nao existe futuro apenas present proximo.
// there is aproblem with comunication of the arduino inputs due to input beeing 8bit word.
//The more in depth the state machine is the more obidiente and dumb it becomes, because
//it will have to specify all possible cases.
//char *X, X is a variable that stores a fisical
//address with a cast type char.
//getnum works with string terminating with only NULL or '\0'
//my style
//Lesson always be flexible and except and obey the good practices,
// without any desobidience, and for extra culture if desired try to
// clarify the why things are as they are. Never be stuborn.
```

```
#include<avr/io.h>
#define TRUE 1
#define FALSE 0
#define column 4
#define lines 36

void setup()
{
 // start serial port at 9600 bps:
 Serial.begin(9600);
 DDRA = B00000000;
 PORTA = B11111111;
 DDRC = B11111111;
 PORTC = B00000000;
}

void loop()
{
 int i, l, c;
 int data[5];
 int Hist[5];
 int count=0;
 //inic key
 data[0]=0;
 data[1]=255;
 data[2]=0;
 data[4]=0;
 //mem prepared for depth 2 in FSM (finie state machine).
 int mem[36][5]=
 {
  { 0, 0, 0,255, 0},
  { 0, 0, 0,254, 1},
  { 0, 0, 1,253, 0},
  { 0, 0, 1,254, 2},
  { 0, 0, 2,254, 1},
  {0 , 0, 1,300, 0}//timer
 };

 /******************CICLOS DE MAQUINA*********************/
 for( Hist[0] = data[0], Hist[1] = data[1], Hist[2] = data[2]; TRUE; Hist[3] = data[3]){

  data[3] = PINA;

  delay(60);


  if(!(Serial.available()>0)){
```

```
  Serial.print(data[3]);
  Serial.print(", ");
  Serial.println(count);
}

//timer_1 setup
if(data[4] == 1){
  count++;
}else{
  count = 0;
}
//catch timer
if(count == 500){
  data[3] = 300;
}
//end timer

/**********************/
if(data[3] == Hist[3])
  continue;
if(data[1] == data[3])
  continue;
/*********************/
/*********************Search and apply changes********************/
for( l = 0, c = 0; l < lines   ; l++){
  if(c >= column){
    l--;
    data[4] = mem[l][c];
    //update
    data[0]=data[2];
    data[1]=data[3];
    data[2]=data[4];
    Hist[0]=data[0];
    Hist[1]=data[1];
    Hist[2]=data[2];
    Hist[4]=data[4];
    //send
    PORTC = data[4];
    break;
  }
  //startup c=2, c=0, for more precise.
  for(c=2; c < column; c++){
    if(data[c] == mem[l][c]){
      continue;
    }else{
      break;
    }
  }
}
```

```
   /*****FALL THREW*************************************************************/







   /**************************************************************************/
  }
}
//Nao existe futuro apenas present proximo.
// there is aproblem with comunication of the arduino inputs due to input beeing 8bit word.
//The more in depth the state machine is the more obidiente and dumb it becomes, because
//it will have to specify all possible cases.
//char *X, X is a variable that stores a fisical
//address with a cast type char.
//getnum works with string terminating with only NULL or '\0'
//my style
//Lesson always be flexible and except and obey the good practices,
// without any desobidience, and for extra culture if desired try to
// clarify the why things are as they are. Never be stuborn.
```

```
#include<avr/io.h>
#include<avr/interrupt.h>
#define TRUE 1
#define FALSE 0
#define column 4
#define lines 52
void setup()
{
  // start serial port at 9600 bps:
  Serial.begin(9600);
  DDRA = B00000000;
  PORTA = B11111111;
  DDRC = B11111111;
  PORTC = B00000000;
}

void loop()
{
  int i, l, c;
  int data[5];
  int Hist[5];
  int count=0;
  //inic key
  data[0]=0;
  data[1]=63;
  data[2]=0;
  data[4]=0;
  //mem prepared for depth 2 in FSM (finie state machine).

 const int mem[(lines+1)][(column+1)]=
  {
    { 0, 0,65,300, 0},//timer off.
    { 0, 0, 0,21, 0},//1
    { 0, 0,65,21,65},//2
    { 0, 0, 0,17,65},//3output 1 and timer on.
    { 0, 0,65,17,65},//4
    { 0, 0, 0,20,65},//5
    { 0, 0,65,20,65},//6
    { 0, 0, 0, 5,65},//7
    { 0, 0,65, 5,65},//8
    { 0, 0,65,29, 0},//9
    { 0, 0, 0,29, 0},//10
    { 0, 0,65,23, 0},//11
    { 0, 0, 0,23, 0},//12
    { 0, 0,65,31, 0},//13
    { 0, 0, 0,31, 0},//14
    { 0, 0,65, 7, 0},//15
    { 0, 0, 0, 7, 0},//16
    { 0, 0,65,13, 0},//17
    { 0, 0, 0,13, 0},//18
    { 0, 0,65,37, 0},//19
    { 0, 0, 0,37, 0},//20
    { 0, 0,65,28, 0},//21
    { 0, 0, 0,28, 0},//22
    { 0, 0,65,25, 0},//23
    { 0, 0, 0,25, 0},//24
    { 0, 0,65,17,65},//25
    { 0, 0, 0,13, 0},//26
    { 0, 0, 0,53, 0},//27
    { 0, 0,65,53, 0},//28
    { 0, 0, 0,53, 0},//29
    { 0, 0,65,22, 0},//30
    { 0, 0, 0,22, 0},//31
    { 0, 0,65,61, 0},//32
    { 0, 0, 0,61, 0},//33
    { 0, 0,65,55, 0},//34
    { 0, 0, 0,55, 0},//35
    { 0, 0,65,63, 0},//36
    { 0, 0, 0,63, 0},//37
    { 0, 0,65,52, 0},//38
    { 0, 0, 0,52, 0},//39
    { 0, 0,65,19, 0},//40
    { 0, 0, 0,19, 0},//41
    { 0, 0,65, 9, 0},//42
    { 0, 0, 0, 9, 0},//43
    { 0, 0,65,12, 0},//44
    { 0, 0, 0,12, 0},//45
    { 0, 0, 0,24, 0},//46
    { 0, 0, 0,24, 0},//47
    { 0, 0,65,49, 0},//48
    { 0, 0, 0,49, 0}//49
  };

  /*******************CICLOS DE MAQUINA*********************/
  for( Hist[0] = data[0], Hist[1] = data[1], Hist[2] = data[2], count = 0; TRUE; Hist[3] = data[3]){


    data[3] = 63 & PINA;

    delay(60);


    if(!(Serial.available()>0)){
      Serial.print(data[3]);
      Serial.print(", ");
      Serial.println(count);
    }

    //timer_1 setup
    if(data[4] == 65){
      count++;
    }else{
      count = 0;
    }
    //catch timer
    if(count == 150){
      data[3] = 300;
      count = 0;
    }
    //end timer

    /**********************/
    if(data[3] == Hist[3])
      continue;
    if(data[1] == data[3])
      continue;
    /*********************/
    /*********************Search and apply changes********************/
    for( l = 0, c = 0; l < lines   ; l++){
      if(c >= column){
        l--;
        data[4] = mem[l][c];
        //update
        data[0]=data[2];
        data[1]=data[3];
        data[2]=data[4];
        Hist[0]=data[0];
        Hist[1]=data[1];
        Hist[2]=data[2];
        Hist[4]=data[4];
        //send
        PORTC = 63 & data[4];
        break;
      }
      //startup c=2, c=0, for more precise.
      for(c=2; c < column; c++){
        if(data[c] == mem[l][c]){
          continue;
        }else{
          break;
        }
      }
    }
    /*****FALL THREW*********************************************************/
```

```
   /**********************************************************************************/
  }
}
//Nao existe futuro apenas present proximo.
// there is aproblem with comunication of the arduino inputs due to input beeing 8bit word.
//The more in depth the state machine is the more obidiente and dumb it becomes, because
//it will have to specify all possible cases.
//char *X, X is a variable that stores a fisical
//address with a cast type char.
//getnum works with string terminating with only NULL or '\0'
//my style
//Lesson always be flexible and except and obey the good practices,
// without any desobidience, and for extra culture if desired try to
// clarify the why things are as they are. Never be stuborn.
//use the pow formula for intvar.
```

```
//Este programa é aplicado para botoneiras start/stop,
//aplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemilanove.
#include<avr/io.h>
#include<avr/interrupt.h>
#define TRUE 1
#define FALSE 0
#define column 2
#define lines 52

void setup()
{
  // start serial port at 9600 bps:
  Serial.begin(9600);
  DDRA = B00000000;
  PORTA = B11111111;
  DDRC = B11111111;
  PORTC = B00000000;
}

void loop()
{
  int i, l, c;
  int data[3];
  int Hist[3];
  int treat;
  int count=0;
  //inic key
  data[0] = 0;
  data[2] = 0;
  //mem prepared for depth 2 in FSM (finite state machine).
  const int mem[(lines+1)][(column+1)]=
  {
    {65,300, 0},//timer off output 0
    { 0,21, 0},//1
    {65,21,65},//2
    { 0,17,65},//3timer on and output 1
    {65,17,65},//4
    { 0,20,65},//5
    {65,20,65},//6
    { 0, 5,65},//7
    {65, 5,65},//8
    {65,29, 0},//9
    { 0,29, 0},//10
    {65,23, 0},//11
    { 0,23, 0},//12
    {65,31, 0},//13
    { 0,31, 0},//14
    {65, 7, 0},//15
    { 0, 7, 0},//16
    {65,13, 0},//17
    { 0,13, 0},//18
    {65,37, 0},//19
    { 0,37, 0},//20
    {65,28, 0},//21
    { 0,28, 0},//22
    {65,25, 0},//23
    { 0,25, 0},//24
    {65,17,65},//25
    { 0,13, 0},//26
    { 0,53, 0},//27
    {65,53, 0},//28
    { 0,53, 0},//29
    {65,22, 0},//30
    { 0,22, 0},//31
    {65,61, 0},//32
    { 0,61, 0},//33
    {65,55, 0},//34
    { 0,55, 0},//35
    {65,63, 0},//36
    { 0,63, 0},//37
    {65,52, 0},//38
    { 0,52, 0},//39
    {65,19, 0},//40
    { 0,19, 0},//41
    {65, 9, 0},//42
    { 0, 9, 0},//43
    {65,12, 0},//44
    { 0,12, 0},//45
    { 0,24, 0},//46
    { 0,24, 0},//47
    {65,49, 0},//48
    { 0,49, 0}//49
  };

  /******************CICLOS DE MAQUINA*********************/
  for( Hist[0] = data[0], Hist[2] = data[2], count = 0; TRUE; Hist[1] = data[1]){

    treat = 63 & PINA;
    data[1] = treat;

    delay(60);

    if(!(Serial.available()>0)){
      Serial.print(data[1]);
      Serial.print(" , ");
      Serial.println(count);
    }

    //timer setup
    if(data[2] == 65){
      count++;
    }else{
      count = 0;
    }
    //catch timer
    if(count == 15000){
      data[1] = 300;
      count = 0;
    }
    //end timer

    /**********************/
    if(data[1] == Hist[1])
      continue;
    //if(!(Serial.available()>0)){
    //  Serial.println(data[1]);
    //}
    /********************/
    /********************Search and apply changes*******************/
    for( l = 0, c = 0; l < lines   ; l++){
      if(c >= column){
        l--;
        data[2] = mem[l][c];
        //update
        data[0] = data[2];
        Hist[0] = data[0];
        //send
        PORTC = 63 & data[2];
        //if(!(Serial.available()>0)){
        //  Serial.println(data[2]);
        //}
        break;
      }
      /***/
      for(c=0; c < column; c++){
        if(data[c] == mem[l][c]){
          continue;
        }else{
          break;
        }
      }
    }
    /************************FALL THREW*************************************/
```

```
    /*************************************************************************/
  }
}
//The least error prone.
```

```
//Este programa é aplicado para botoneiras start/stop,
//aplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemilanove.
#include<avr/io.h>
#include<avr/interrupt.h>
#define TRUE 1
#define FALSE 0
#define column 2
#define lines 52

void setup()
{
 // start serial port at 9600 bps:
 Serial.begin(9600);
 DDRA = B00000000;
 PORTA = B11111111;
 DDRC = B11111111;
 PORTC = B00000000;
}

void loop()
{
 int i, l, c;
 int data[3];
 int Hist[3];
 int treat;
 int count_1=0,count_2=0;
 //inic key
 data[0] = 0;
 data[2] = 0;
 //mem prepared for depth 2 in FSM (finite state machine).
 const int mem[(lines+1)][(column+1)]=
 {
  {0,254,257},//timer off output 0
  {257,254,0},//1
  {256,254,0},//2
  {257,300,256},//3timer on and output 1
  {256,400,257},//4
  {0,253,1},//5
  {1,253,0},//6
  { 0, 5,65},//7
  {65, 5,65},//8
  {65,29, 0},//9
  { 0,29, 0},//10
  {65,23, 0},//11
  { 0,23, 0},//12
  {65,31, 0},//13
```

```
   { 0,31, 0},//14
   {65, 7, 0},//15
   { 0, 7, 0},//16
   {65,13, 0},//17
   { 0,13, 0},//18
   {65,37, 0},//19
   { 0,37, 0},//20
   {65,28, 0},//21
   { 0,28, 0},//22
   {65,25, 0},//23
   { 0,25, 0},//24
   {65,17,65}//25
  };

  /*******************CICLOS DE MAQUINA*********************/
  for( Hist[0] = data[0], Hist[2] = data[2], count_1 = 0, count_2 = 0; TRUE; Hist[1] = data[1]){

    treat = 255 & PINA;
    data[1] = treat;

    delay(60);

    //timer_1
    if(data[2] == 257){
      count_1++;
    }else{
      count_1 = 0;
    }
    //catch timer
    if(count_1 == 10){
      data[1] = 300;
      count_1 = 0;
    }
    //end timer
 //timer_2
    if(data[2] == 256){
      count_2++;
    }else{
      count_2 = 0;
    }
    //catch timer
    if(count_2 == 10){
      data[1] = 400;
      count_2 = 0;
    }
    //end timer

    /*********************/
    if(data[1] == Hist[1])
```

```
      continue;
    if(!(Serial.available()>0)){
      Serial.println(data[1]);
    }
    /*********************/
/*********************Search and apply changes********************/
    for( l = 0, c = 0; l < lines   ; l++){
      if(c >= column){
        l--;
        data[2] = mem[l][c];
        //update
        data[0] = data[2];
        Hist[0] = data[0];
        //send
        PORTC = 255 & data[2];
        if(!(Serial.available()>0)){
          Serial.println(data[2]);
        }
        break;
      }
      /***/
      for(c=0; c < column; c++){
        if(data[c] == mem[l][c]){
          continue;
        }else{
          break;
        }
      }
    }
/*************************FALL THREW*************************************/




    /*************************************************************************/
  }
}
//The least error prone.
```

```
//Este programa é aplicado para botoneiras start/stop,
//aplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemilanove.
#include<avr/io.h>
#include<avr/interrupt.h>
#define TRUE 1
#define FALSE 0
#define column 2
#define lines 52

void setup()
{
 // start serial port at 9600 bps:
 //Serial.begin(9600);
 DDRA = B00000000;
 PORTA = B11111111;
 DDRC = B11111111;
 PORTC = B00000000;
}

void loop()
{
 int i, l, c;
 int data[3];
 int Hist[3];
 int treat;
 int count_1 = 0,count_2 = 0;
 //inic key
 data[0] = 0;
 data[2] = 0;
 //mem prepared for depth 2 in FSM (finite state machine).
 const int mem[(lines+1)][(column+1)]=
 {
   {0,254,257},//
   {257,254,0},//1
   {0,253,1},//5
   {1,253,0}//6
 };

 /*******************CICLOS DE MAQUINA*********************/
 for( Hist[0] = data[0], Hist[2] = data[2], count_1 = 0, count_2 = 0; TRUE; Hist[1] = data[1]){

   treat = 255 & PINA;
   data[1] = treat;

   delay(60);
```

```
//timer_1
    if(data[2] == 257){
      count_1++;
    }else{
      count_1 = 0;
    }
    //catch timer
    if(count_1 == 6){
      data[2] = 256;
      PORTC = 255 & data[2];
      count_1 = 0;
    }
    //end timer
//timer_2
    if(data[2] == 256){
      count_2++;
    }else{
      count_2 = 0;
    }
    //catch timer
    if(count_2 == 6){
      data[2] = 257;
      PORTC = 255 & data[2];
      count_2 = 0;
    }
    //end timer

    /**********************/
    if(data[1] == Hist[1])
      continue;
    //if(!(Serial.available()>0)){
      //Serial.println(data[1]);
    //}
    /********************/
/********************Search and apply changes********************/
    for( l = 0, c = 0; l < lines   ; l++){
      if(c >= column){
        l--;
        data[2] = mem[l][c];
        //update
        data[0] = data[2];
        Hist[0] = data[0];
        Hist[1] = data[1];
        //send
        PORTC = 255 & data[2];
        break;
      }
      /***/
      for(c=0; c < column; c++){
        if(data[c] == mem[l][c]){
          continue;
        }else{
          break;
        }
      }
    }
/**************************FALL THREW***************************************/




/******************************************************************************/
  }
}
//The least error prone.
```

```
//Este programa é aplicado para botoneiras start/stop,
//aplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemilanove.
#include<avr/io.h>
#include<avr/interrupt.h>
#define TRUE 1
#define FALSE 0
#define column 2
#define lines 52

void setup()
{
  // start serial port at 9600 bps:
  Serial.begin(9600);
  DDRA = B00000000;
  PORTA = B11111111;
  DDRC = B11111111;
  PORTC = B00000000;
}

void loop()
{
  int i, l, c;
  int data[3];
  int Hist[2];
  int treat;
  int count_1 = 0,count_2 = 0;
  //inic key
  data[0] = 0;
  data[2] = 0;
  //mem prepared for depth 2 in FSM (finite state machine).
  const int mem[(lines+1)][(column+1)]=
  {
    {0,254,257},//1
    {257,254,0},
    {256,254,0},//2
    {257,300,256},
    {256,400,257},
    {0,253,1},//3
    {1,253,0}//4

  };

  /*******************CICLOS DE MAQUINA*********************/
  for( Hist[0] = data[0], count_1 = 0, count_2 = 0; TRUE; Hist[1] = data[1]){

    treat = 255 & PINA;
    data[1] = treat;

    delay(60);

//timer_1
    if(data[2] == 257){
      count_1++;
    }else{
      count_1 = 0;
    }
    //catch timer
    if(count_1 == 100){
      data[1] = 300;
      count_1 = 0;
    }
    //end timer
//timer_2
    if(data[2] == 256){
      count_2++;
    }else{
      count_2 = 0;
    }
    //catch timer
    if(count_2 == 100){
      data[1] = 400;
      count_2 = 0;
    }
    //end timer

    /**********************/
    if(data[1] == Hist[1])
      continue;
    if(!(Serial.available()>0)){
      Serial.println(data[1]);
    }
    /*********************/
/*********************Search and apply changes********************/
    for( l = 0, c = 0; l < lines   ; l++){
      if(c >= column){
        l--;
        data[2] = mem[l][c];
        //update
        data[0] = data[2];
        if(data[1] > 255)
          data[1] = Hist[1];
        break;
      }
      /***/
      for(c=0; c < column; c++){
        if(data[c] == mem[l][c]){
          continue;
        }else{
          break;
        }
      }
    }
/*************************FALL THREW**************************************/
    PORTC = 255 & data[2];




/************************************************************************/
  }
}
//The least error prone.
```

```
//Este programa é aplicado para botoneiras start/stop,
//aplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemilanove.
#include<avr/io.h>
#include<avr/interrupt.h>
#define TRUE 1
#define FALSE 0
#define column 2
#define lines 52

void setup()
{
  // start serial port at 9600 bps:
  Serial.begin(9600);
  DDRA = B00000000;
  PORTA = B11111111;
  DDRC = B11111111;
  PORTC = B00000000;
}

void loop()
{
  int i, l, c;
  int data[3];
  int Hist[2];
  int treat;
  int count_1 = 0,count_2 = 0;
  //inic key
  data[0] = 0;
  data[2] = 0;
  //mem prepared for depth 2 in FSM (finite state machine).
  const int mem[(lines+1)][(column+1)]=
  {
    {0,254,257},//1
    {257,254,0},
    {256,254,0},//2
    {257,300,256},
    {256,400,257},
    {0,253,1},//3
    {1,253,0}//4

  };

  /******************CICLOS DE MAQUINA*********************/
  for( Hist[0] = data[0], count_1 = 0, count_2 = 0; TRUE; Hist[1] = data[1]){

    treat = 255 & PINA;
    data[1] = treat;

    delay(60);

//timer_1
    if(data[2] == 257){
      count_1++;
    }else{
      count_1 = 0;
    }
    //catch timer
    if(count_1 == 100){
      data[1] = 300;
      count_1 = 0;
    }
    //end timer
//timer_2
    if(data[2] == 256){
      count_2++;
    }else{
      count_2 = 0;
    }
    //catch timer
    if(count_2 == 100){
      data[1] = 400;
      count_2 = 0;
    }
    //end timer

    /*********************/
    if(data[1] == Hist[1])
      continue;
    if(!(Serial.available()>0)){
      Serial.println(data[1]);
    }
    /********************/
/********************Search and apply changes*******************/
    for( l = 0, c = 0; l < lines   ; l++){
      if(c >= column){
        l--;
        data[2] = mem[l][c];
        //update
        data[0] = data[2];
        if(data[1] > 255)
          data[1] = Hist[1];
        break;
      }
      /***/
      for(c=0; c < column; c++){
        if(data[c] == mem[l][c]){
          continue;
        }else{
          break;
        }
      }
    }
/************************FALL THREW*************************************/
    PORTC = 255 & data[2];




/**********************************************************************/
  }
}
//The least error prone.
```

```
//Este programa é aplicado para botoneiras start/stop,
//aplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemilanove.
#include<avr/io.h>
#include<avr/interrupt.h>
#define TRUE 1
#define FALSE 0
#define column 2
#define lines 52

void setup()
{
  // start serial port at 9600 bps:
  Serial.begin(9600);
  DDRA = B00000000;
  PORTA = B11111111;
  DDRC = B11111111;
  PORTC = B00000000;
}

void loop()
{
  int i, l, c;
  int data[3];
  int Hist[2];
  int treat;
  int count_1 = 0,count_2 = 0;
  //inic key
  data[0] = 0;
  data[2] = 0;
  //mem prepared for depth 2 in FSM (finite state machine).
    const int mem[(lines+1)][(column+1)]=
  {
    {65,300, 0},//timer off output 0
    { 0,21, 0},//1
    {65,21,65},//2
    { 0,17,65},//3timer on and output 1
    {65,17,65},//4
    { 0,20,65},//5
    {65,20,65},//6
    { 0, 5,65},//7
    {65, 5,65},//8
    {65,29, 0},//9
    { 0,29, 0},//10
    {65,23, 0},//11
    { 0,23, 0},//12
    {65,31, 0},//13
    { 0,31, 0},//14
    {65, 7, 0},//15
    { 0, 7, 0},//16
    {65,13, 0},//17
    { 0,13, 0},//18
    {65,37, 0},//19
    { 0,37, 0},//20
    {65,28, 0},//21
    { 0,28, 0},//22
    {65,25, 0},//23
    { 0,25, 0},//24
    {65,17,65},//25
    { 0,13, 0},//26
    { 0,53, 0},//27
    {65,53, 0},//28
    { 0,53, 0},//29
    {65,22, 0},//30
    { 0,22, 0},//31
    {65,61, 0},//32
    { 0,61, 0},//33
    {65,55, 0},//34
    { 0,55, 0},//35
    {65,63, 0},//36
    { 0,63, 0},//37
    {65,52, 0},//38
    { 0,52, 0},//39
    {65,19, 0},//40
    { 0,19, 0},//41
    {65, 9, 0},//42
    { 0, 9, 0},//43
    {65,12, 0},//44
    { 0,12, 0},//45
    { 0,24, 0},//46
    { 0,24, 0},//47
    {65,49, 0},//48
    { 0,49, 0}//49
  };


  /*******************CICLOS DE MAQUINA*********************/
  for( Hist[0] = data[0], count_1 = 0, count_2 = 0; TRUE; Hist[1] = data[1]){

    treat = 63 & PINA;
    data[1] = treat;

    delay(60);

//timer_1
    if(data[2] == 65){
      count_1++;
    }else{
      count_1 = 0;
    }
    //catch timer
    if(count_1 == 100){
      data[1] = 300;
      count_1 = 0;
    }
    //end timer
//timer_2
    if(data[2] == 64){
      count_2++;
    }else{
      count_2 = 0;
    }
    //catch timer
    if(count_2 == 100){
      data[1] = 400;
      count_2 = 0;
    }
    //end timer
    /**********************/
    if(data[1] == Hist[1])
      continue;
    if(!(Serial.available()>0)){
      Serial.println(data[1]);
    }
    /********************/
/*********************Search and apply changes********************/
    for( l = 0, c = 0; l < lines   ; l++){
      if(c >= column){
        l--;
        data[2] = mem[l][c];
        //update
        data[0] = data[2];
        if(data[1] > 63)
          data[1] = Hist[1];
        break;
      }
      /***/
      for(c=0; c < column; c++){
        if(data[c] == mem[l][c]){
          continue;
        }else{
          break;
        }
      }
    }
```

```
/************************FALL THREW************************************/
   PORTC = 63 & data[2];




/**********************************************************************/
  }
}
//The least error prone.
```

```
//Este programa é aplicado para botoneiras start/stop,
//aplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemialnove.
#include<avr/io.h>
#include<avr/interrupt.h>
#define TRUE 1
#define FALSE 0
#define column 2
#define lines 52

void setup()
{
  // start serial port at 9600 bps:
  //Serial.begin(9600);
  DDRC = B00000000;
  PORTC = B11111111;
  DDRB = B11111111;
  PORTB = B00000000;
}

void loop()
{
  int i, l, c;
  int data[3];
  int Hist[2];
  int treat;
  int count_1 = 0;
  int timon=0;
  int timoff=99;
  //inic key
  data[0] = 0;
  data[2] = 0;
  //mem prepared for depth 2 in FSM (finite state machine).
  const int mem[(lines+1)][(column+1)]=
  {
    {0,62,65},//timer off output 0
    {65,300,64},//1
    {64,400,65},//2
    {65,62,0},//3timer on and output 1
    {64,62,0}//4

  };

/*******************CICLOS DE MAQUINA*********************/
  for( Hist[0] = data[0], count_1 = 0; TRUE; Hist[1] = data[1]){

    treat = 63 & PINC;
    data[1] = treat;

    delayMicroseconds(250);


    if((data[2] == 65) || (data[2] == 64)){
      count_1++;
      if(count_1 == 100)
        count_1=0;
      if(count_1 == timoff)
        data[1]=300;
      if(count_1 == timon)
        data[1]=400;
    }

    //if(!(Serial.available()>0)){
      //Serial.print(data[1]);
      //Serial.print(" , ");
      //Serial.print(count_1);
      //Serial.print(" , ");
      //Serial.print(timon);
      //Serial.print(" , ");
      //Serial.println(timoff);
    //}

    /**********************/
    if(data[1] == Hist[1])
      continue;

    /********************/
/*****************FSM****Search and apply changes********************/
    for( l = 0, c = 0; (l < lines) && (c < column); l++){
      for(c=0; c < column; c++){
        if(data[c] == mem[l][c]){
          continue;
        }else{
          break;
        }
      }
    }
    if(c == column){
        l--;
        data[2] = mem[l][c];
        //update
        data[0] = data[2];
    }
/************************FALL THREW*************************************/
    PORTB = 63 & data[2];
    /******************/
```

```
    if(data[1] > 63){
          data[1] = Hist[1];
          continue;
    }
    /*********LOGIC**********/
    if((data[1] == 61) && (timon < 49)){
      timon++;
      timoff--;
    }
    if((data[1] == 59) && (timon > 0) ){
      timoff++;
      timon--;
    }




/***************************************************************************/
  }
}
//The least error prone.
```

```
//Este programa é aplicado para botoneiras start/stop,
//aplicado em minha casa. Vamos ver duração
//teste endurance de placa arduino duemialnove.
#include<avr/io.h>
#include<avr/interrupt.h>
#define TRUE 1
#define FALSE 0
#define column 2
#define lines 24

void setup()
{
  // start serial port at 9600 bps:
  //Serial.begin(9600);
  DDRC = B00000000;
  PORTC = B11111111;
  DDRB = B11111111;
  PORTB = B00000000;
}

void loop()
{
  int i, l, c;
  int data[3];
  int Hist[2];
  int treat;
  int count_1 = 0;
  int timon=0;
  int timoff=99;
  //inic key
  data[0] = 0;
  data[2] = 0;
  //mem prepared for depth 2 in FSM (finite state machine).
  const int mem[(lines+1)][(column+1)]=
  {
    {0,62,64},//timer off output 0
    {64,300,65},//1 65 ad 64 func
    {65,400,64},//2
    {64,62,0},//3timer on and output 1
    {65,62,0},//4
    {0,31,1},
    {1,31,0}

  };

/*******************CICLOS DE MAQUINA*********************/
  for( Hist[0] = data[0], count_1 = 0; TRUE; Hist[1] = data[1]){
    treat = 63 & PINC;
    data[1] = treat;

    delayMicroseconds(100);


    if((data[2] == 65) || (data[2] == 64)){
      count_1++;
      if(count_1 == 100)
        count_1=0;
      if(count_1 == timoff)
        data[1]=300;
      if(count_1 == timon)
        data[1]=400;
    }else{count_1=0;}

    //if(!(Serial.available()>0)){
      //Serial.print(data[1]);
      //Serial.print(" , ");
      //Serial.print(count_1);
      //Serial.print(" , ");
      //Serial.print(timon);
      //Serial.print(" , ");
      //Serial.println(timoff);
    //}

    /*********************/
    if(data[1] == Hist[1])
      continue;

    /********************/
/****************FSM****Search and apply changes*******************/
    for( l = 0, c = 0; (l < lines) && (c < column); l++){
      for(c=0; c < column; c++){
        if(data[c] == mem[l][c]){
          continue;
        }else{
          break;
        }
      }
    }
    if(c == column){
        l--;
        data[2] = mem[l][c];
        //update
        data[0] = data[2];
    }
/************************FALL THREW*********************************/
```

```
    PORTB = 63 & data[2];
    /*******************/
    if(data[1] > 63){
          data[1] = Hist[1];
          continue;
    }
    /*********LOGIC**********/
    if((data[1] == 61) && (timon < 49)){
      timon++;
      timoff--;
    }
    if((data[1] == 59) && (timon > 0) ){
      timoff++;
      timon--;
    }




/***********************************************************************/
  }
}
//The least error prone.
```