Control de acceso 1.0

Generado por Doxygen 1.11.0

| 1 Índice de archivos                         | 1      |
|--|--------|
| 1.1 Lista de archivos                        | <br>1  |
| 2 Documentación de archivos                  | 3      |
| 2.1 Referencia del archivo main.cpp          | <br>3  |
| 2.1.1 Descripción detallada                  | <br>4  |
| 2.1.2 Documentación de funciones             | <br>4  |
| 2.1.2.1 accessDeniedLed()                    | <br>4  |
| 2.1.2.2 accessGrantedLed()                   | <br>4  |
| 2.1.2.3 keypadCols()                         | <br>5  |
| 2.1.2.4 keypadRows()                         | <br>5  |
| 2.1.2.5 main()                               | <br>5  |
| 2.1.2.6 uartUSB()                            | <br>6  |
| 2.1.2.7 workingLed()                         | <br>6  |
| 2.1.3 Documentación de variables             | <br>6  |
| 2.1.3.1 accessState                          | <br>6  |
| 2.1.3.2 accumulatedTime                      | <br>6  |
| 2.1.3.3 colReading                           | <br>6  |
| 2.1.3.4 currentIndex                         | <br>6  |
| 2.1.3.5 inputSequence                        | <br>7  |
| 2.1.3.6 keypadAllowedEntries                 | <br>7  |
| 2.1.3.7 rowReading                           | <br>7  |
| 2.1.3.8 sequenceDetectedState                | <br>7  |
| 2.1.3.9 startTimeout                         | <br>7  |
| 2.1.3.10 timeoutState                        | <br>7  |
| 2.2 main.cpp                                 | <br>8  |
| 2.3 Referencia del archivo mis_funciones.cpp | <br>9  |
| 2.3.1 Descripción detallada                  | <br>9  |
| 2.3.2 Documentación de funciones             | <br>10 |
| 2.3.2.1 areEqual()                           | <br>10 |
| 2.3.2.2 clearSequence()                      | <br>10 |
| 2.3.2.3 displayAccessDeniedMsg()             | <br>10 |
| 2.3.2.4 displayAccessGrantedMsg()            | <br>10 |
| 2.3.2.5 displayTimeoutMsg()                  | <br>11 |
| 2.3.2.6 getKeyPressed()                      | <br>11 |
| 2.3.2.7 keypadInit()                         | <br>11 |
| 2.3.2.8 keypadSweepUpdate()                  | <br>12 |
| 2.3.2.9 uartInit()                           | <br>12 |
| 2.4 mis_funciones.cpp                        | <br>12 |
| 2.5 Referencia del archivo mis_funciones.hpp | <br>14 |
| 2.5.1 Descripción detallada                  | 15     |
| 2.5.2 Documentación de «define»              | <br>15 |

| 2.5.2.1 ACCESS_SEQUENCE           | . 15 |
|-----------------------------------|------|
| 2.5.2.2 BLINKING_TIME             | . 15 |
| 2.5.2.3 COL_LEN                   | . 15 |
| 2.5.2.4 ROW_LEN                   | . 15 |
| 2.5.2.5 SEQUENCE_LEN              | . 16 |
| 2.5.2.6 TIMEOUT                   | . 16 |
| 2.5.2.7 USERNAME                  | . 16 |
| 2.5.3 Documentación de funciones  | . 16 |
| 2.5.3.1 areEqual()                | . 16 |
| 2.5.3.2 clearSequence()           | . 16 |
| 2.5.3.3 displayAccessDeniedMsg()  | . 17 |
| 2.5.3.4 displayAccessGrantedMsg() | . 17 |
| 2.5.3.5 displayTimeoutMsg()       | . 17 |
| 2.5.3.6 getKeyPressed()           | . 17 |
| 2.5.3.7 keypadInit()              | . 18 |
| 2.5.3.8 keypadSweepUpdate()       | . 18 |
| 2.5.3.9 uartInit()                | . 18 |
| 2.6 mis_funciones.hpp             | . 19 |
| dice alfabético                   | 21   |

## Capítulo 1

# Índice de archivos

## 1.1. Lista de archivos

Lista de todos los archivos con breves descripciones:

| main.cpp          |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | <br> |  | 3  |
|-------------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|------|--|----|
| mis_funciones.cpp |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      |  | Ç  |
| mis funciones hon |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |      |  | 14 |

2 Índice de archivos

## Capítulo 2

## Documentación de archivos

## 2.1. Referencia del archivo main.cpp

```
#include "mbed.h"
#include <cstdint>
#include <string.h>
#include "arm_book_lib.h"
#include "mis funciones.hpp"
```

## **Funciones**

■ BusOut keypadRows (PC 6, PC 7, PC 8, PC 9)

Declaración de GPIOs asociados a las filas del keypad.

BusIn keypadCols (PC\_0, PC\_1, PC\_2, PC\_3)

Declaración de GPIOs asociados a las Columnas del keypad.

DigitalOut workingLed (PA 5)

Declaración de GPIO asociado a LED que indica funcionamiento del sistema.

DigitalOut accessGrantedLed (PA\_6)

Declaración de GPIO asociado a LED que indica acceso concedido.

■ DigitalOut accessDeniedLed (PA\_7)

Declaración de GPIO asociado a LED que indica acceso denegado.

UnbufferedSerial uartUSB (USBTX, USBRX)

Declaración de UART para mostrar los mensajes del control de acceso.

■ int main ()

## **Variables**

bool sequenceDetectedState

Flag indicador de secuencia completa. Flag en true indica secuencia completa.

bool accessState

Flag indicador de estado de acceso. Flag en true indica acceso concedido.

bool timeoutState

Flag indicador de timeout. Flag en true indica timeout alcanzado.

bool startTimeout

int colReading

Respaldo del estado de las columnas del keypad luego de detectar tecla presionada.

int rowReading

Respaldo del estado de las filas del keypad luego de detectar tecla presionada.

uint8\_t currentIndex

Indice actual de la secuencia.

uint16\_t accumulatedTime

Tiempo acumulado utilizado para comparar con tiempo de timeout.

char inputSequence [SEQUENCE\_LEN+1]

Array utilizado para guardar la secuencia de caracteres ingresados.

char keypadAllowedEntries [ROW\_LEN][COL\_LEN]

Array que contiene los caracteres asociados a las teclas del keypad.

## 2.1.1. Descripción detallada

Versión

1.0

Fecha

18/7/24

**Autor** 

Ramos Leonardo

Definición en el archivo main.cpp.

## 2.1.2. Documentación de funciones

## 2.1.2.1. accessDeniedLed()

```
DigitalOut accessDeniedLed ( PA_7)
```

Declaración de GPIO asociado a LED que indica acceso denegado.

## 2.1.2.2. accessGrantedLed()

```
DigitalOut accessGrantedLed ( $\operatorname{\mathtt{PA\_6}} )
```

Declaración de GPIO asociado a LED que indica acceso concedido.

## 2.1.2.3. keypadCols()

```
BusIn keypadCols (  \begin{array}{c} PC\_0 \ , \\ PC\_1 \ , \\ PC\_2 \ , \\ PC\_3 \ ) \end{array}
```

Declaración de GPIOs asociados a las Columnas del keypad.

## 2.1.2.4. keypadRows()

```
BusOut keypadRows ( PC_6 , PC_7 , PC_8 , PC_9 )
```

Declaración de GPIOs asociados a las filas del keypad.

## 2.1.2.5. main()

```
int main ()
```

Definición en la línea 35 del archivo main.cpp.

```
keypadInit();
        uartInit();
workingLed = 0;
00038
00039
00040
        while (true) {
  if (accumulatedTime < TIMEOUT) {</pre>
00041
             keypadSweepUpdate();
00042
00043
             if (sequenceDetectedState == true) {
00044
              if (areEqual()) {
00045
                 accessState = true;
00046
              } else {
00047
                 accessState = false;
00048
00049
               if (accessState == true) {
00050
                displayAccessGrantedMsg();
00051
                 accessGrantedLed = ON;
00052
                 delay(3000);
                 accessGrantedLed = OFF;
accessState = false;
00053
00054
00055
               } else {
00056
                displayAccessDeniedMsg();
00057
                 accessDeniedLed = ON;
00058
                 delay(1000);
00059
                 accessDeniedLed = OFF;
00060
00061
              clearSequence();
00062
               sequenceDetectedState = false;
00063
               timeoutState = false;
00064
               accumulatedTime = 0;
00065
00066
             delay(10);
00067
             accumulatedTime += 10;
00068
00069
00070
               if(timeoutState == true){
00071
                   displayTimeoutMsg();
00072
                   clearSequence();
timeoutState = false;
00073
00074
00075
               accumulatedTime = 0;
00076
           if((accumulatedTime% BLINKING_TIME) == 0){
00077
00078
               workingLed = !workingLed;
00079
08000
        }
00082 }
```

## 2.1.2.6. uartUSB()

Declaración de UART para mostrar los mensajes del control de acceso.

## 2.1.2.7. workingLed()

```
DigitalOut workingLed ( PA_5 )
```

Declaración de GPIO asociado a LED que indica funcionamiento del sistema.

### 2.1.3. Documentación de variables

### 2.1.3.1. accessState

```
bool accessState
```

Flag indicador de estado de acceso. Flag en true indica acceso concedido.

Definición en la línea 22 del archivo main.cpp.

### 2.1.3.2. accumulatedTime

```
uint16_t accumulatedTime
```

Tiempo acumulado utilizado para comparar con tiempo de timeout.

Definición en la línea 28 del archivo main.cpp.

## 2.1.3.3. colReading

```
int colReading
```

Respaldo del estado de las columnas del keypad luego de detectar tecla presionada.

Definición en la línea 25 del archivo main.cpp.

## 2.1.3.4. currentIndex

```
uint8_t currentIndex
```

Indice actual de la secuencia.

Definición en la línea 27 del archivo main.cpp.

## 2.1.3.5. inputSequence

```
char inputSequence[SEQUENCE_LEN+1]
```

Array utilizado para guardar la secuencia de caracteres ingresados.

Definición en la línea 29 del archivo main.cpp.

## 2.1.3.6. keypadAllowedEntries

```
char keypadAllowedEntries[ROW_LEN][COL_LEN]
```

#### Valor inicial:

```
= {{'1', '2', '3', 'A'},
```

Array que contiene los caracteres asociados a las teclas del keypad.

Definición en la línea 30 del archivo main.cpp.

```
{{'1', '2', '3', 'A'}, 
{'4', '5', '6', 'B'}, 
{'7', '8', '9', 'C'}, 
{'*', '0', '#', 'D'}};
00031
00032
00033
```

#### 2.1.3.7. rowReading

```
int rowReading
```

Respaldo del estado de las filas del keypad luego de detectar tecla presionada.

Definición en la línea 26 del archivo main.cpp.

## 2.1.3.8. sequenceDetectedState

```
bool sequenceDetectedState
```

Flag indicador de secuencia completa. Flag en true indica secuencia completa.

Definición en la línea 21 del archivo main.cpp.

### 2.1.3.9. startTimeout

```
bool startTimeout
```

Definición en la línea 24 del archivo main.cpp.

## 2.1.3.10. timeoutState

```
bool timeoutState
```

Flag indicador de timeout. Flag en true indica timeout alcanzado.

Definición en la línea 23 del archivo main.cpp.

## 2.2. main.cpp

## Ir a la documentación de este archivo.

```
00001
00007 #include "mbed.h"
00008 #include <cstdint>
00009 #include <string.h>
00010 #include "arm_book_lib.h"
00011 #include "mis_funciones.hpp"
00012
00013 // main() runs in its own thread in the OS
00014 BusOut keypadRows(PC_6, PC_7, PC_8, PC_9);
00015 BusIn keypadCols(PC_0, PC_1, PC_2, PC_3);
00016 DigitalOut workingLed(PA_5);
00017 DigitalOut accessGrantedLed(PA_6);
00018 DigitalOut accessDeniedLed(PA_7);
00019 UnbufferedSerial uartUSB(USBTX, USBRX);
00020
00021 bool sequenceDetectedState;
00022 bool accessState;
00023 bool timeoutState;
00024 bool startTimeout;
00025 int colReading;
00026 int rowReading;
00027 uint8_t currentIndex;
00028 uint16_t accumulatedTime;
00029 char inputSequence[SEQUENCE_LEN + 1];
00029 char inputSequence[SEQUENCE_LEN + 1];

00030 char keypadAllowedEntries[ROW_LEN][COL_LEN] = {{'1', '2', '3', 'A'},

00031 {'4', '5', '6', 'B'},

00032 {'7', '8', '9', 'C'},

00033 {'*', '0', '#', 'D'}};
00034
00035 int main() {
00037 keypadInit();
        uartInit();
workingLed = 0;
00038
00039
00040
        while (true) {
         if (accumulatedTime < TIMEOUT) {</pre>
00041
00042
            keypadSweepUpdate();
00043
             if (sequenceDetectedState == true) {
00044
              if (areEqual()) {
00045
                 accessState = true;
00046
               } else {
                 accessState = false;
00048
00049
               if (accessState == true)
00050
               displayAccessGrantedMsg();
00051
                 accessGrantedLed = ON;
00052
                 delay(3000);
                 accessGrantedLed = OFF;
00053
00054
                 accessState = false;
00055
               displayAccessDeniedMsg();
00056
00057
                  accessDeniedLed = ON;
                  delay(1000);
00058
00059
                 accessDeniedLed = OFF;
00060
00061
               clearSequence();
00062
               sequenceDetectedState = false;
00063
               timeoutState = false;
00064
               accumulatedTime = 0;
00065
             delay(10);
00067
             accumulatedTime += 10;
00068
00069
           elsef
               if(timeoutState == true){
00070
00071
                   displayTimeoutMsg();
00072
                    clearSequence();
00073
                    timeoutState = false;
00074
               accumulatedTime = 0;
00075
00076
00077
           if((accumulatedTime% BLINKING_TIME) == 0){
00078
               workingLed = !workingLed;
00080 }
00082 }
```

## 2.3. Referencia del archivo mis funciones.cpp

#include "mis\_funciones.hpp"

#### **Funciones**

void keypadInit ()

keypadInit Inicializa el keypad 4x4 para su uso. Principalmente configura los GPIO de columnas como input PULL DOWN y da valores iniciales a ciertos flags.

void uartInit ()

uartInit Configura velocidad de transmisión y formato de la trama de datos de UART

void keypadSweepUpdate ()

keypadSweepUpdate Barre el keypad 4x4 hasta encontrar una tecla presionada, guarda el caracter asociado a esa tecla e indica mediante flag sequenceDetectedState que se ha ingresado la secuencia completa

char getKeyPressed ()

getKeyPressed Realiza la conversion de posicion en keypad (fila y columna) a caracter asociado

bool areEqual ()

areEqual Compara igualdad elemento a elemento entre el array de la secuencia ingresada y el string de usuario guardado en memoria

void displayAccessGrantedMsg ()

displayAccessGrantedMsg Muestra mensajes asociados a un acceso concedido en terminal

void displayAccessDeniedMsg ()

displayAccessDeniedMsg Muestra mensajes asociados a un acceso denegado en terminal

void displayTimeoutMsg ()

displayTimeoutMsg Muestra mensaje de timeout en terminal

void clearSequence ()

clearSequence Llena de ceros el array que recibe la secuencia desde el keypad y reinicia su índice.

## 2.3.1. Descripción detallada

Versión

1.0

Fecha

18/7/24

Autor

Ramos Leonardo

Definición en el archivo mis\_funciones.cpp.

### 2.3.2. Documentación de funciones

## 2.3.2.1. areEqual()

```
bool areEqual ()
```

areEqual Compara igualdad elemento a elemento entre el array de la secuencia ingresada y el string de usuario guardado en memoria

### Devuelve

true si son iguales, false si no lo son.

Definición en la línea 111 del archivo mis funciones.cpp.

#### 2.3.2.2. clearSequence()

```
void clearSequence ()
```

clearSequence Llena de ceros el array que recibe la secuencia desde el keypad y reinicia su índice.

Definición en la línea 153 del archivo mis\_funciones.cpp.

#### 2.3.2.3. displayAccessDeniedMsg()

```
\verb"void displayAccessDeniedMsg" ()
```

displayAccessDeniedMsg Muestra mensajes asociados a un acceso denegado en terminal

Definición en la línea 134 del archivo mis\_funciones.cpp.

## 2.3.2.4. displayAccessGrantedMsg()

```
void displayAccessGrantedMsg ()
```

displayAccessGrantedMsg Muestra mensajes asociados a un acceso concedido en terminal

Definición en la línea 122 del archivo mis funciones.cpp.

## 2.3.2.5. displayTimeoutMsg()

```
void displayTimeoutMsg ()
```

displayTimeoutMsg Muestra mensaje de timeout en terminal

Definición en la línea 145 del archivo mis funciones.cpp.

## 2.3.2.6. getKeyPressed()

```
char getKeyPressed ()
```

getKeyPressed Realiza la conversion de posicion en keypad (fila y columna) a caracter asociado

#### Devuelve

Caracter asociado a la tecla presionada

Definición en la línea 87 del archivo mis\_funciones.cpp.

```
00087
        uint8_t currentRow;
00089
        uint8_t currentCol;
        for (uint8_t i = 0; i < ROW_LEN; i++) {
  if ((rowReading & (1 « i)) != 0) {</pre>
00090
00091
00092
            currentRow = i;
00093
             i = ROW_LEN;
00094
         }
00095
00096
        for (uint8_t i = 0; i < COL_LEN; i++) {</pre>
        if ((colReading & (1 « i)) != 0) {
00097
00098
          currentCol = i;
00099
             i = COL LEN:
00100
          }
00101
00102
        uartUSB.write("*", 1);
00103
        return keypadAllowedEntries[currentRow][currentCol];
00104 }
```

## 2.3.2.7. keypadlnit()

```
void keypadInit ()
```

keypadInit Inicializa el keypad 4x4 para su uso. Principalmente configura los GPIO de columnas como input PULL DOWN y da valores iniciales a ciertos flags.

Definición en la línea 33 del archivo mis funciones.cpp.

## 2.3.2.8. keypadSweepUpdate()

```
void keypadSweepUpdate ()
```

keypadSweepUpdate Barre el keypad 4x4 hasta encontrar una tecla presionada, guarda el caracter asociado a esa tecla e indica mediante flag sequenceDetectedState que se ha ingresado la secuencia completa

Definición en la línea 58 del archivo mis\_funciones.cpp.

```
00058
00059
         for (uint8_t i = 0; i < ROW_LEN; i++) {</pre>
          keypadRows = (1 « i);
00060
00061
           if
              (keypadCols != 0) {
00062
             delay(100);
             if (keypadCols != 0) {
  timeoutState = true;
00063
00064
00065
               accumulatedTime = 0:
               rowReading = keypadRows.read();
colReading = keypadCols.read();
00066
00067
00068
                inputSequence[currentIndex] = getKeyPressed();
00069
               currentIndex++;
               if (currentIndex >= SEQUENCE_LEN) {
00070
00071
                 uartUSB.write("\n", 1);
00072
                  sequenceDetectedState = true;
00074
                i = ROW_LEN;
00075
00076
          }
00077
00078
        keypadRows = 0;
        return;
00080 }
```

#### 2.3.2.9. uartInit()

```
void uartInit ()
```

uartInit Configura velocidad de transmisión y formato de la trama de datos de UART

Definición en la línea 45 del archivo mis funciones.cpp.

```
00045 {
00046 uartUSB.baud(9600);
00047 uartUSB.format(/*bits*/ 8,
00048 /*paridad*/ SerialBase::None,
00049 /*bits de stop*/ 1);
00050 }
```

## 2.4. mis\_funciones.cpp

Ir a la documentación de este archivo.

```
00007 #include "mis_funciones.hpp"
80000
00010 extern BusOut keypadRows;
00011 extern BusIn keypadCols;
00012 extern DigitalOut workingLed;
00013 extern DigitalOut accessGrantedLed;
00014 extern DigitalOut accessDeniedLed;
00015 extern UnbufferedSerial uartUSB;
00016
00017 extern bool sequenceDetectedState;
00018 extern bool accessState:
00019 extern bool timeoutState;
00020 extern int colReading;
00021 extern int rowReading;
00022 extern uint8_t currentIndex;
00023 extern uint16_t accumulatedTime;
00024 extern char inputSequence[SEQUENCE_LEN + 1];
00025 extern char keypadAllowedEntries[ROW_LEN][COL_LEN];
00033 void keypadInit() {
```

```
keypadCols.mode(PullDown);
00035
         inputSequence[SEQUENCE_LEN] = '\0';
00036
         sequenceDetectedState = false;
00037
         currentIndex = 0;
00038
        accumulatedTime = 0;
00039 }
00045 void uartInit() {
00046 uartUSB.baud(9600);
00047
        uartUSB.format(/*bits*/ 8,
                          /*paridad*/ SerialBase::None,
00048
00049
                          /*bits de stop*/ 1);
00050 }
00051
00058 void keypadSweepUpdate() {
        for (uint8_t i = 0; i < ROW_LEN; i++) {
  keypadRows = (1 « i);</pre>
00059
00060
              (keypadCols != 0) {
00061
           if
00062
             delay(100);
00063
              if (keypadCols != 0) {
00064
                timeoutState = true;
00065
                accumulatedTime = 0;
                rowReading = keypadRows.read();
colReading = keypadCols.read();
00066
00067
00068
                inputSequence[currentIndex] = getKeyPressed();
00069
                currentIndex++;
00070
                if (currentIndex >= SEQUENCE_LEN) {
                  uartUSB.write("\n", 1);
00071
00072
                  sequenceDetectedState = true;
00073
00074
                i = ROW LEN:
00075
             }
00076
00077
00078
         keypadRows = 0;
00079
         return;
00080 }
00087 char getKeyPressed() {
00088
        uint8_t currentRow;
00089
         uint8_t currentCol;
         for (uint8_t i = 0; i < ROW_LEN; i++) {
   if ((rowReading & (1 « i)) != 0) {
      currentRow = i;
   }</pre>
00090
00091
00092
00093
             i = ROW_LEN;
00094
00095
         for (uint8_t i = 0; i < COL_LEN; i++) {</pre>
00096
00097
           if ((colReading & (1 « i)) != 0) {
00098
             currentCol = i;
00099
              i = COL_LEN;
00100
00101
00102
         uartUSB.write("*", 1);
         return keypadAllowedEntries[currentRow][currentCol];
00103
00104 }
00111 bool areEqual(){
00112
           bool equalState = true;
00113
           if(strcmp(inputSequence, ACCESS_SEQUENCE) != 0){
                equalState = false;
00114
00115
00116
           return equalState;
00117 }
00118
00122 void displayAccessGrantedMsg() {
00123    uartUSB.write("Acceso concedido\n", 17);
         uartUSB.write("Usuario:", 8);
00124
        uartUSB.write(USERNAME, strlen(USERNAME));
uartUSB.write("Clave:", 6);
uartUSB.write(ACCESS_SEQUENCE, strlen(ACCESS_SEQUENCE));
00125
00126
00127
00128
        uartUSB.write("\n", 1);
00129 }
00130
00134 void displayAccessDeniedMsg() {
00135 uartUSB.write("Acceso denegado\n", 16);
00136
         uartUSB.write("Usuario desconocido\n", 20);
00137
         uartUSB.write("Clave:", 6);
        uartUSB.write(inputSequence, SEQUENCE_LEN);
uartUSB.write("\n", 1);
00138
00139
00140 }
00141
00145 void displayTimeoutMsg(){
           uartUSB.write("\nTimeout!!!\n", 12);
00146
00147 }
00148
00153 void clearSequence(){
```

## 2.5. Referencia del archivo mis funciones.hpp

```
#include "mbed.h"
#include <cstdint>
#include <string.h>
#include "arm book lib.h"
```

#### defines

■ #define ROW LEN 4

Cantidad de filas del keypad.

#define COL\_LEN 4

Cantidad de columnas del keypad.

■ #define SEQUENCE LEN 4

Largo de la secuencia ingresada.

#define ACCESS\_SEQUENCE "ABCD"

Secuencia de acceso guardada en memoria.

■ #define USERNAME "Leonardo\n"

Usuario asociado a la secuencia de acceso en memoria.

■ #define TIMEOUT 5000

Tiempo de timeout en ms.

■ #define BLINKING TIME 500

Tiempo de parpadeo (ms) del led de funcionamiento del sistema.

## **Funciones**

void keypadInit ()

keypadInit Inicializa el keypad 4x4 para su uso. Principalmente configura los GPIO de columnas como input PULL DOWN y da valores iniciales a ciertos flags.

void uartInit ()

uartInit Configura velocidad de transmisión y formato de la trama de datos de UART

void keypadSweepUpdate ()

keypadSweepUpdate Barre el keypad 4x4 hasta encontrar una tecla presionada, guarda el caracter asociado a esa tecla e indica mediante flag sequenceDetectedState que se ha ingresado la secuencia completa

char getKeyPressed ()

getKeyPressed Realiza la conversion de posicion en keypad (fila y columna) a caracter asociado

void displayAccessGrantedMsg ()

displayAccessGrantedMsg Muestra mensajes asociados a un acceso concedido en terminal

void displayAccessDeniedMsg ()

displayAccessDeniedMsg Muestra mensajes asociados a un acceso denegado en terminal

void displayTimeoutMsg ()

displayTimeoutMsg Muestra mensaje de timeout en terminal

bool areEqual ()

areEqual Compara igualdad elemento a elemento entre el array de la secuencia ingresada y el string de usuario guardado en memoria

void clearSequence ()

clearSequence Llena de ceros el array que recibe la secuencia desde el keypad y reinicia su índice.

## 2.5.1. Descripción detallada

Versión

1.0

Fecha

18/7/24

**Autor** 

Ramos Leonardo

Definición en el archivo mis\_funciones.hpp.

## 2.5.2. Documentación de «define»

## 2.5.2.1. ACCESS\_SEQUENCE

```
#define ACCESS_SEQUENCE "ABCD"
```

Secuencia de acceso guardada en memoria.

Definición en la línea 18 del archivo mis\_funciones.hpp.

## 2.5.2.2. BLINKING\_TIME

```
#define BLINKING_TIME 500
```

Tiempo de parpadeo (ms) del led de funcionamiento del sistema.

Definición en la línea 21 del archivo mis\_funciones.hpp.

## 2.5.2.3. COL\_LEN

```
#define COL_LEN 4
```

Cantidad de columnas del keypad.

Definición en la línea 16 del archivo mis\_funciones.hpp.

## 2.5.2.4. ROW\_LEN

```
#define ROW_LEN 4
```

Cantidad de filas del keypad.

Definición en la línea 15 del archivo mis\_funciones.hpp.

## 2.5.2.5. SEQUENCE\_LEN

```
#define SEQUENCE_LEN 4
```

Largo de la secuencia ingresada.

Definición en la línea 17 del archivo mis\_funciones.hpp.

#### 2.5.2.6. TIMEOUT

```
#define TIMEOUT 5000
```

Tiempo de timeout en ms.

Definición en la línea 20 del archivo mis\_funciones.hpp.

### 2.5.2.7. **USERNAME**

```
#define USERNAME "Leonardo\n"
```

Usuario asociado a la secuencia de acceso en memoria.

Definición en la línea 19 del archivo mis\_funciones.hpp.

### 2.5.3. Documentación de funciones

## 2.5.3.1. areEqual()

```
bool areEqual ()
```

areEqual Compara igualdad elemento a elemento entre el array de la secuencia ingresada y el string de usuario guardado en memoria

Devuelve

true si son iguales, false si no lo son.

Definición en la línea 111 del archivo mis\_funciones.cpp.

## 2.5.3.2. clearSequence()

```
void clearSequence ()
```

clearSequence Llena de ceros el array que recibe la secuencia desde el keypad y reinicia su índice.

Definición en la línea 153 del archivo mis\_funciones.cpp.

#### 2.5.3.3. displayAccessDeniedMsg()

```
void displayAccessDeniedMsg ()
```

displayAccessDeniedMsg Muestra mensajes asociados a un acceso denegado en terminal

Definición en la línea 134 del archivo mis\_funciones.cpp.

### 2.5.3.4. displayAccessGrantedMsg()

```
void displayAccessGrantedMsg ()
```

displayAccessGrantedMsg Muestra mensajes asociados a un acceso concedido en terminal

Definición en la línea 122 del archivo mis funciones.cpp.

### 2.5.3.5. displayTimeoutMsg()

```
void displayTimeoutMsg ()
```

displayTimeoutMsg Muestra mensaje de timeout en terminal

Definición en la línea 145 del archivo mis\_funciones.cpp.

#### 2.5.3.6. getKeyPressed()

```
char getKeyPressed ()
```

getKeyPressed Realiza la conversion de posicion en keypad (fila y columna) a caracter asociado

Devuelve

Caracter asociado a la tecla presionada

Definición en la línea 87 del archivo mis\_funciones.cpp.

```
00087
00088
        uint8_t currentRow;
00089
        uint8_t currentCol;
        for (uint8_t i = 0; i < ROW_LEN; i++) {</pre>
00090
         if ((rowReading & (1 « i)) != 0) {
00091
00092
            currentRow = i;
00093
            i = ROW_LEN;
00094
         }
00095
        for (uint8_t i = 0; i < COL_LEN; i++) {</pre>
00096
        if ((colReading & (1 « i)) != 0) {
00097
00098
           currentCol = i;
00099
            i = COL_LEN;
00100
         }
00101
00102
        uartUSB.write("*", 1);
00103
        return keypadAllowedEntries[currentRow][currentCol];
00104 }
```

## 2.5.3.7. keypadlnit()

```
void keypadInit ()
```

keypadInit Inicializa el keypad 4x4 para su uso. Principalmente configura los GPIO de columnas como input PULL DOWN y da valores iniciales a ciertos flags.

Definición en la línea 33 del archivo mis\_funciones.cpp.

## 2.5.3.8. keypadSweepUpdate()

```
void keypadSweepUpdate ()
```

keypadSweepUpdate Barre el keypad 4x4 hasta encontrar una tecla presionada, guarda el caracter asociado a esa tecla e indica mediante flag sequenceDetectedState que se ha ingresado la secuencia completa

Definición en la línea 58 del archivo mis\_funciones.cpp.

```
00058
00059
         for (uint8_t i = 0; i < ROW_LEN; i++) {</pre>
00060
          keypadRows = (1 « i);
00061
              (keypadCols != 0) {
           if
00062
             delay(100);
00063
             if (keypadCols != 0) {
00064
               timeoutState = true;
               accumulatedTime = 0;
00065
               rowReading = keypadRows.read();
colReading = keypadCols.read();
00066
00067
00068
               inputSequence[currentIndex] = getKeyPressed();
00069
               currentIndex++;
00070
               if (currentIndex >= SEQUENCE_LEN) {
                 uartUSB.write("\n", 1);
00071
00072
                 sequenceDetectedState = true;
00073
00074
               i = ROW_LEN;
00075
             }
00076
          }
00077
00078
        keypadRows = 0;
00079
        return;
00080 }
```

## 2.5.3.9. uartInit()

```
void uartInit ()
```

uartInit Configura velocidad de transmisión y formato de la trama de datos de UART

Definición en la línea 45 del archivo mis\_funciones.cpp.

## 2.6. mis\_funciones.hpp

## Ir a la documentación de este archivo.

```
00001
00007 #ifndef MIS_FUNCIONES_H
00008 #define MIS_FUNCIONES_H
00010 #include "mbed.h"
00011 #include <cstdint>
00012 #include <string.h>
00013 #include "arm_book_lib.h"
00014
00015 #define ROW_LEN 4
00016 #define COL_LEN 4
00017 #define SEQUENCE_LEN 4
00018 #define ACCESS_SEQUENCE "ABCD"
00019 #define USERNAME "Leonardo\n"
00020 #define TIMEOUT 5000
00021 #define BLINKING_TIME 500
00023 /*Defino prototipos de funciones*/
00024 void keypadInit();
00025 void uartInit();
00025 void tartific(),
00026 void keypadSweepUpdate();
00027 char getKeyPressed();
00028 void displayAccessGrantedMsg();
00029 void displayAccessDeniedMsg();
00030 void displayTimeoutMsg();
00031 bool areEqual();
00032 void clearSequence();
00033
00034 #endif // MIS_FUNCIONES_H
```

# Índice alfabético

| ACCESS_SEQUENCE                  | mis_funciones.hpp, 17       |
|----------------------------------|-----------------------------|
| mis_funciones.hpp, 15            | keypadRows                  |
| accessDeniedLed                  | main.cpp, 5                 |
| main.cpp, 4                      | keypadSweepUpdate           |
| accessGrantedLed                 | mis_funciones.cpp, 11       |
| main.cpp, 4                      | mis_funciones.hpp, 18       |
| accessState                      |                             |
| main.cpp, 6                      | main                        |
| accumulatedTime                  | main.cpp, 5                 |
| main.cpp, 6                      | main.cpp, 3                 |
| areEqual                         | accessDeniedLed, 4          |
| mis_funciones.cpp, 10            | accessGrantedLed, 4         |
| mis_funciones.hpp, 16            | accessState, 6              |
|                                  | accumulatedTime, 6          |
| BLINKING_TIME                    | colReading, 6               |
| mis_funciones.hpp, 15            | currentIndex, 6             |
|                                  | inputSequence, 6            |
| clearSequence                    | keypadAllowedEntries, 7     |
| mis_funciones.cpp, 10            | keypadCols, 4               |
| mis_funciones.hpp, 16            | keypadRows, 5               |
| COL_LEN                          | main, 5                     |
| mis_funciones.hpp, 15            | rowReading, 7               |
| colReading                       | sequenceDetectedState, 7    |
| main.cpp, 6                      | startTimeout, 7             |
| currentIndex                     | timeoutState, 7             |
| main.cpp, 6                      | uartUSB, 5                  |
|                                  | workingLed, 6               |
| displayAccessDeniedMsg           | mis_funciones.cpp, 9        |
| mis_funciones.cpp, 10            | areEqual, 10                |
| mis_funciones.hpp, 16            | clearSequence, 10           |
| displayAccessGrantedMsg          | displayAccessDeniedMsg, 10  |
| mis_funciones.cpp, 10            | displayAccessGrantedMsg, 10 |
| mis_funciones.hpp, 17            | displayTimeoutMsg, 10       |
| displayTimeoutMsg                | getKeyPressed, 11           |
| mis_funciones.cpp, 10            | keypadInit, 11              |
| mis_funciones.hpp, 17            | keypadSweepUpdate, 11       |
|                                  | uartInit, 12                |
| getKeyPressed                    | mis_funciones.hpp, 14       |
| mis_funciones.cpp, 11            | ACCESS_SEQUENCE, 15         |
| mis_funciones.hpp, 17            | areEqual, 16                |
|                                  | BLINKING_TIME, 15           |
| inputSequence                    | clearSequence, 16           |
| main.cpp, 6                      | COL_LEN, 15                 |
| lances and Allances all Catalana | displayAccessDeniedMsg, 16  |
| keypadAllowedEntries             | displayAccessGrantedMsg, 17 |
| main.cpp, 7                      | displayTimeoutMsg, 17       |
| keypadCols                       | getKeyPressed, 17           |
| main.cpp, 4                      | keypadInit, 17              |
| keypadInit                       | keypadSweepUpdate, 18       |
| mis funciones con 11             | reypausweepupuale, 18       |

22 ÍNDICE ALFABÉTICO

| ROW_LEN, 15<br>SEQUENCE_LEN, 15<br>TIMEOUT, 16<br>uartInit, 18<br>USERNAME, 16 |
|--|
| ROW_LEN  |
| mis_funciones.hpp, 15  |
| rowReading   |
| main.cpp, 7  |
| SEQUENCE LEN   |
| mis_funciones.hpp, 15  |
| sequenceDetectedState  |
| main.cpp, 7  |
| startTimeout   |
| main.cpp, 7  |
| TIMEOUT  |
| mis_funciones.hpp, 16  |
| timeoutState   |
| main.cpp, 7  |
| uartInit   |
| mis_funciones.cpp, 12  |
| mis_funciones.hpp, 18  |
| uartUSB  |
| main.cpp, 5  |
| USERNAME   |
| mis_funciones.hpp, 16  |
| workingLed   |
| main.cpp, 6  |
|  |