```c
/* Keypad.c - implemetation of Keypad.h
*
*     Copyright 2020 Graeme Judge, Sean Berkvens
*   Change Log:
*       May 3, 2020: Source file created
*/


#include "Keypad.h"

#define KEYDELAY for(int i =0; i < 8000; i++)

#define DELLY for(int i =0; i < 100000; i++)

uint8_t rows[4] = {ROW1, ROW2, ROW3, ROW4};

void GPIOInit(){
  GPIOInitRow();
  GPIOInitCol();

}

void GPIOInitRow(){
  //Clock init
  RCC -> AHB2ENR |= RCC_AHB2ENR_GPIOEEN;
  //using PE12 to PE15
  for(int i = 12; i < 16; i++){ //for loop to go through all of the pins being used and initializes them
    // GPIO Mode
    //00 input mode
    //01 output mode
    //10 alternate mode
    //11 analog mode
    GPIOE->MODER &= ~(3UL<<(2*i));
    GPIOE->MODER |= (1UL<<(2*i));       //  Output(01)

    //GPIO Speed
    GPIOE->OSPEEDR &= (~(3UL<<(2*i)));
    GPIOE->OSPEEDR |=   (3UL<<(2*i));  // Low speed

    //00 none
    //01 pullup
    //10 pulldown
    //11 reserved
    GPIOE->PUPDR   |= (1UL<<(2*i));  // pull-up

    //GPIO Output Type: Output push-pull (0, reset), Output open drain (1)
    GPIOE->OTYPER |= (1U<<(1*i));       // Push-pull open drain
  }

  //clears the output data register for port E
  GPIOE->ODR &= (uint32_t)0x0000;
}



void GPIOInitCol(){
  //Clock init
  RCC -> AHB2ENR |= RCC_AHB2ENR_GPIOAEN;

  //using PA1 to PA5 excluding 4
  for(int i = 1; i < 4; i++){ //for loop to go through all of the pins being used and initializes them
      // GPIO Mode
      //00 input mode
      //01 output mode
      //10 alternate mode
      //11 analog mode
      GPIOA->MODER &= ~(3UL<<(2*i));

      //00 none
      //01 pullup
      //10 pulldown
      //11 reserved
```

```
73          GPIOA->PUPDR |= (1UL<<(2*i));  // pull-up
74        }
75
76          // GPIO Pin 5
77          //00 input mode
78          //01 output mode
79          //10 alternate mode
80          //11 analog mode
81          GPIOA->MODER &= ~(3UL<<(2*5));
82
83          //00 none
84          //01 pullup
85          //10 pulldown
86          //11 reserved
87          GPIOA->PUPDR |= (1UL<<(2*5));  // pull-up
88    }
89
90
91
92    enum Keys scan(){
93      GPIOE->ODR &= 0b0000 << 12;
94      DELLY;
95      uint8_t button = debouncedKey();
96
97      if((button & 0xFF) == 0x2E){
98        return Key_None;
99      }
100
101     for(int i = 0; i < 4; i++){
102       GPIOE->ODR = (rows[i])<<12;
103       DELLY;
104       button = debouncedKey();
105       if(button != 0x2E){
106         if(button == COL1){
107           return Matrix[i][0];
108         }
109         if(button == COL2){
110           return Matrix[i][1];
111         }
112         if(button == COL3){
113           return Matrix[i][2];
114         }
115         if(button == COL4){
116           return Matrix[i][3];
117         }
118       }
119     }
120   }
121
122   uint8_t getInput(){
123     volatile uint8_t IDR = (GPIOA->IDR & 0x2E);
124     return IDR;
125   }
126
127   uint8_t debouncedKey(){
128     uint8_t read = getInput();
129
130     for(int i = 0; i < DEBOUNCETIME; i++){
131       uint8_t newRead = getInput();
132       if(read != newRead || read == 0x2E){
133         read = 0x2E;
134         return read;
135       }
136       KEYDELAY;
137     }
138     return read;
139   }
140
141
142
143
144
```

```
145
146
147
```