Embedded System Library Guide

(KEYPAD Library)

Last Modified Date:

Thursday, October 24, 2019

Code author: Majid Derhambakhsh

GitHub: Majid-Derhambakhsh

E-mail

Contents

Guide

To use this library, first add the c file to the program and include the following header.

```
#include "KEYPAD.h"
```

Keypad Configuration

Follow the steps below to set up your library:

1. First open the file **KEYPAD_CONFIG.h**:

   Edit the following values to adjust the connected port used and its pins:

```
#define _KEYPAD_PORT        GPIOx
#define _KEYPAD_PIN         GPIOx
#define _KEYPAD_FIRST_ROW   x
#define _KEYPAD_SCAN_SPEED  x
```

   -- Set the output register: where x can be (A..G depending on device used) to select the GPIO peripheral.

      In AVR: PORTx
      In STM32 ARM: GPIOx

   -- Set the input register: where x can be (A..G depending on device used) to select the GPIO peripheral.

      In AVR: PINx
      In STM32 ARM: GPIOx

   -- Set the first row pin number: This parameter can be 0~(Number of port pins – 8).

   -- Adjust the scan speed of each row (minimum value is 1 millisecond)

Functions

Keypad_GetKey

Function name	uint8_t Keypad_GetKey(void)
Function description	This function is used to get the pushed key number.
Parameters	-
Return values	• Pushed key number (between 0~15).
Notes	-

Example:

```
my_key = Keypad_GetKey();
```

Keypad_Read

Function name	*void* Keypad_Read(*CommandFunction_T* your_function);
Function description	This function is used to get the pushed key number with the release algorithm and run user commands.
Parameters	• **your_function:** User function for run it.
Return values	-
Notes	To use it you have to submit your code as a function, the input function must be in the form void FunctionName(int8_t pushed_key).

Example:

```
void MyFunction(int8_t x)
{
    /* ---- x is the pushed key ---- */
    if(x == 6)
    {
        LED_ON();
    }
    else
    {
        LED_OFF();
    }
}

Keypad_Read(MyFunction);
```

- ```
  void MyFunction(int8_t x)
  {
      /* ---- x is the pushed key ---- */
      sprintf(str, "%u", x);     /* cast pushed key number to string */
      Lcd_PutString(str);      /* show pushed key in the LCD */
  }

  Keypad_Read(MyFunction);
  ```

- ```
  /* Keypad Character */
  uint8_t key_character[16] = {'1' , '2' , '3' , 'A' , '4' , '5' , '6', 'B' , '7' , '8' , '9' , 'C' , '*' , '0' , '#' , 'D'};

  void ShowKey(int8_t ch) /* Function for get key and show on LCD */
  {
      /* ---- x is the pushed key ---- */
      Lcd_PutChar(key_character[ch]);     /* Show character */
  }

  Keypad_Read(ShowKey);
  ```

Requirement

- GPIO driver for AVR microcontrollers.
- HAL & STM32_GPIO drivers for ARM microcontrollers STM32 series.

Important tips

- All commands and settings begin with _.
- All functions are written as Camel Case.
- The functions and codes used in all microcontrollers are the same.

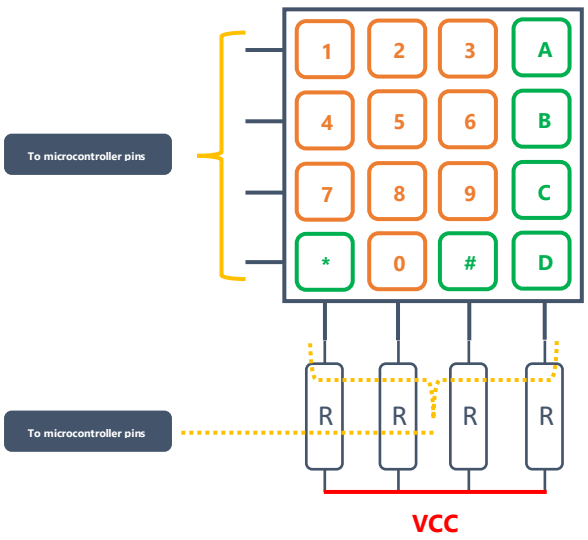
Error & Warning's

Error's:

- Chip or GPIO Library not supported: This error occurs when the microcontroller or its library not supported.

Warning's:

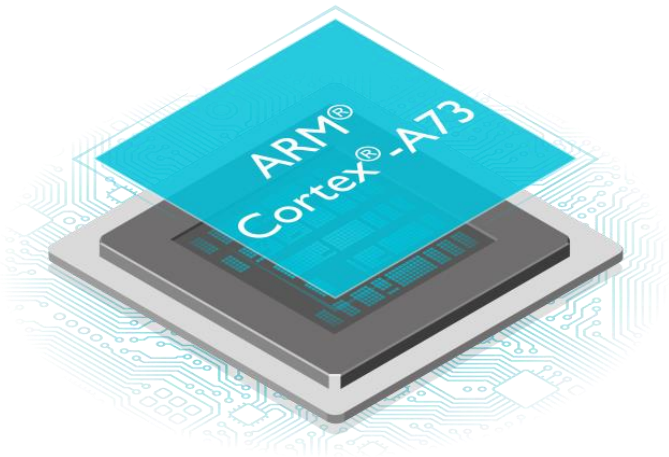
Required Circuit



Supported microcontroller's

ATmega & ATtiny series of AVR Microcontroller's with i2c_unit & gpio_unit driver's

Codevision and GNUC compilers such as AtmelStudio



All STM32 series of ARM Microcontroller's with HAL & stm32_i2c driver's

All ARM Compiler's

Version history

Version 0.0.0

Stable and tested version
