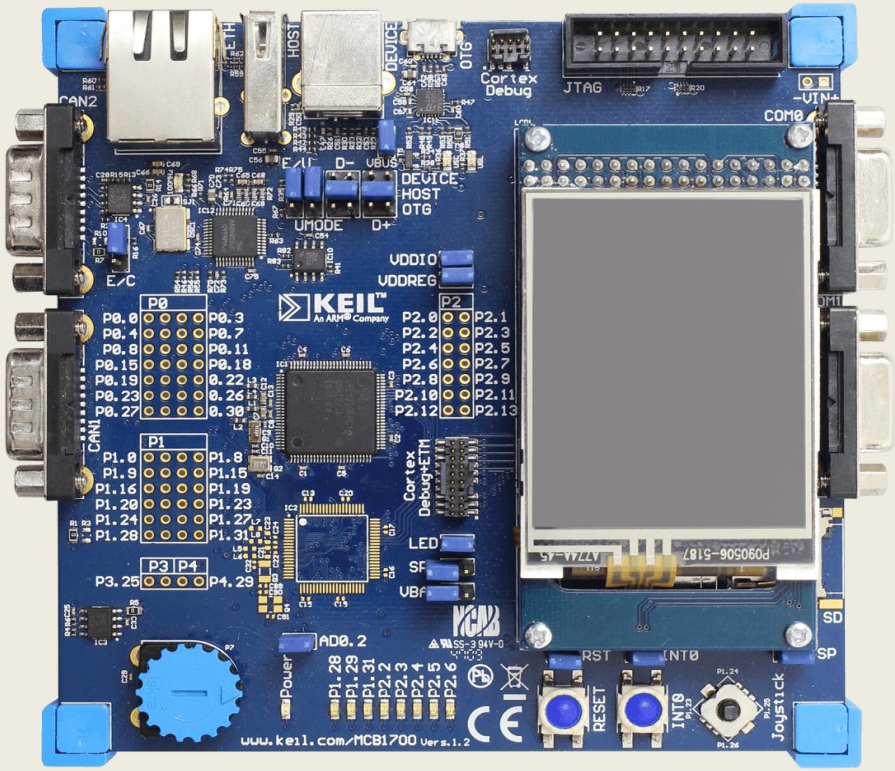
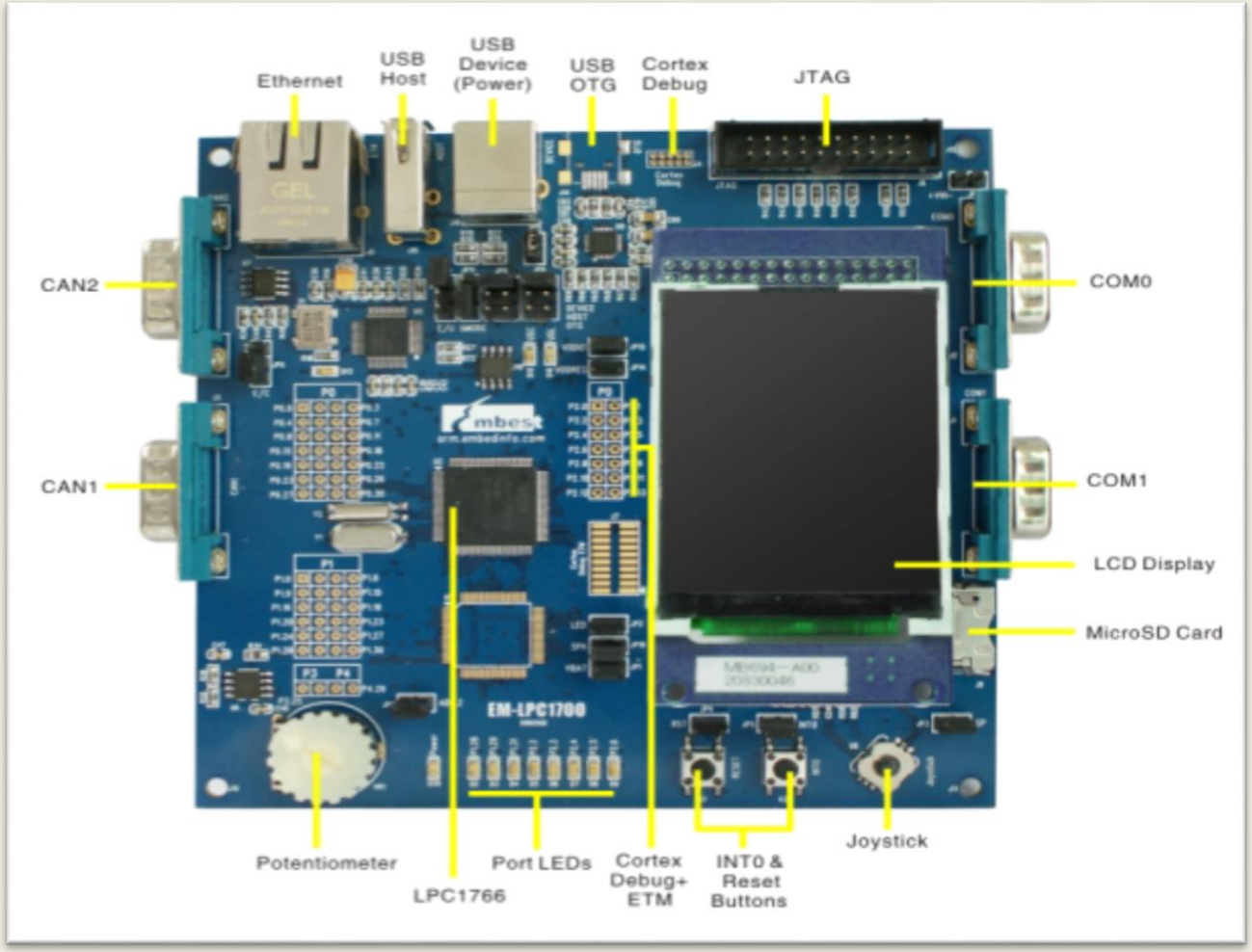
The Keil MCB1700 Evaluation Board



Content

1. Overview
2. GPIO output
3. GPIO input
4. Library
5. Button
6. Joystick
7. LCD
8. ADC

1.1 Board overview



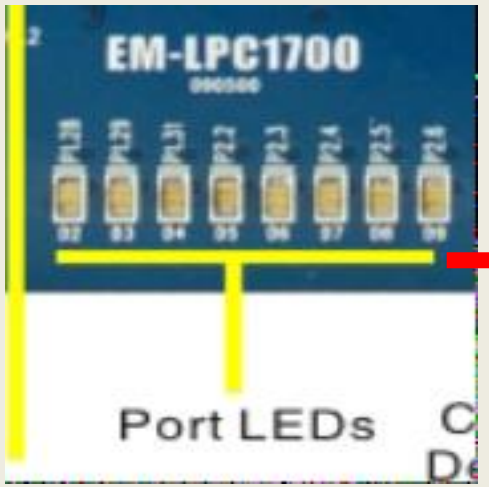
- The frequency up to 100MHz
- Internal Memory: 512KB flash memory & 64KB SRAM
- RS232 & CAN Interfaces
- Ethernet, JTAG & ETM Interface
- LCD Display
- USB Interface
- Analog Output & Input
- Mini SD Card Interface

1.2 Evaluation Board

#	Pin	Type	IO Assignment
1	P1.28 , P1.29, P1.31, P2.2, P2.3, P2.4, P2.5, P2.6	I/O	LEDs
2	P2.10	I/O	Button (EINT0)
3	P1.20 , P1.23 , P1.24 , P1.25, P1.26	I/O	Joystick
4	RSTOUT, P0.8, P0.7, P0.6, P0.5 , P4.28		LCD
5	P0.25	I/O	ADC reading

The two power supply modes below through JP12 configuration.
1) Supply 5V DC through power jack (JP11) on the board.
2) Supply power through USB connecter (CON1) on the board, the current should be less than 500mA.

2.GPIO output



P1.28
P1.29
P1.31
P2.2
P2.3
P2.4
P2.5
P2.6

Port: 1
Pin: 28

Using the function in ***GPIO_LPC17xx.h:***

```
/**
  \fn          void GPIO_PinWrite (uint32_t port_num,
                                   uint32_t pin_num,
                                   uint32_t val);
  \brief       Write port pin
  \param[in]   port_num   GPIO number (0..4)
  \param[in]   pin_num    Port pin number
  \param[in]   val        Port pin value (0 or 1)
*/
extern void GPIO_PinWrite (uint32_t port_num,
                           uint32_t pin_num,
                           uint32_t val);
```

3.GPIO input

- P1.20
- P1.23
- P1.24
- P1.25
- P1.26

Reading the status of Joystick:

```
GPIO_PinRead(uint32_t port_num, uint32_t pin_num);
```

4. Library

- Creating 2 files as name: ***Led_OnOff.c*** and ***Led_OnOff.h***

main.c | Led_OnOff.h | Led_OnOff.c

```c
#include "GPIO_LPC17xx.h"
#include "Led_OnOff.h"

void Led_On( int index)
  {
    switch(index)
      {
      case 1:
        GPIO_PinWrite (1, 28, 1);
      break;
      case 2:
        GPIO_PinWrite (1, 29, 1);
      break;
      case 3:
```

main.c | Led_OnOff.h | Led_OnOff.c

```c
void Led_On(int);

void Led_Off(int);

```

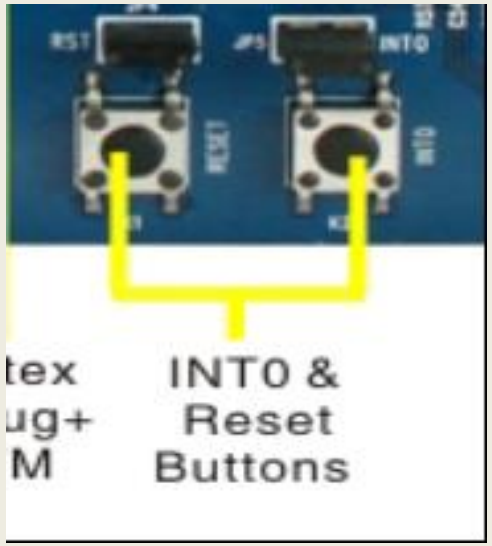
Run:

```c
  int i;

  // Turn ON
  for(i=1;i<=9;i++)
  {
    Led_On(i);
    osDelay(1000);
  }
```

Similar to turn LEDs off

5. Button



There are 2 ways:

1.Reading the GPIO input: P2.10

2.Using the library: *Buttons_MBC1700.c*

```
87 /**
88    \fn          uint32_t Buttons_GetState (void)
89    \brief       Get buttons state
90    \returns     Buttons state
91 */
92 uint32_t Buttons_GetState (void) {
93   uint32_t val;
94
95   val = 0;
96   if (!(GPIO_PinRead (BUTTON_PIN[0].Portnum, BUTTON_PIN[0].Pinnum))) val |= BUTTON_INT0;
97
98   return val;
99 }
```

6. Joystick

There are 2 ways:

- **Reading the GPIO input:**

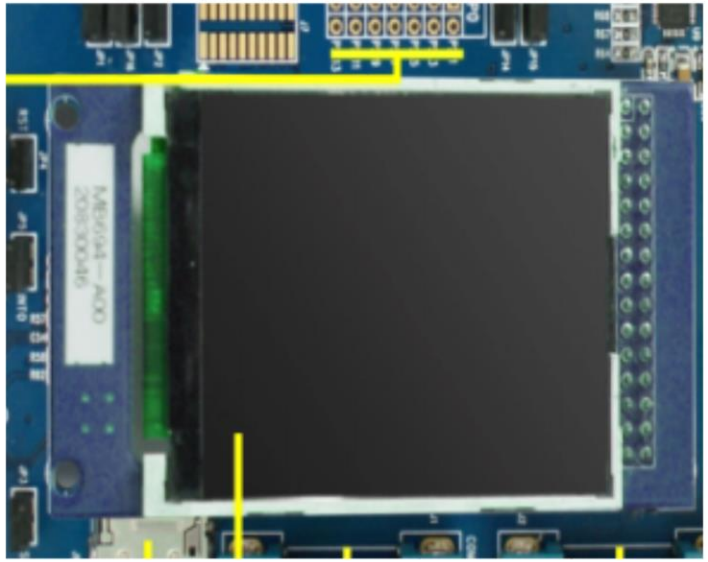
  P1.20 , P1.23 , P1.24 , P1.25, P1.26

- **Using the library: *Board_Joystick.h***

main.c | Board_Joystick.h

```
36
37  #include <stdint.h>
38
39  #define JOYSTICK_LEFT                (1 << 0)  /// Defines the Left-button
40  #define JOYSTICK_RIGHT               (1 << 1)  /// Defines the Right-button
41  #define JOYSTICK_CENTER              (1 << 2)  /// Defines the Center-button
42  #define JOYSTICK_UP                  (1 << 3)  /// Defines the Up-button
43  #define JOYSTICK_DOWN                (1 << 4)  /// Defines the Down-button
44
45  /**
```

7. LCD



2.4 inches TFT LCD（240*320）

- Basing **GLCD_MBC1700.c** to write Code

main.c | GLCD_MCB1700.c

```
1  /*----------------------------------------------------------------------------
2   * Name:    GLCD_MCB1700.c
3   * Purpose: Graphic LCD interface (240x320 pixels) for Graphic LCD with
4   *          SPI interface for MCB1700 evaluation board
5   * Rev.:    1.0.1
6   *----------------------------------------------------------------------------
7
8  /* Copyright (c) 2013 - 2014 ARM LIMITED
```

- Configurating SPI as:

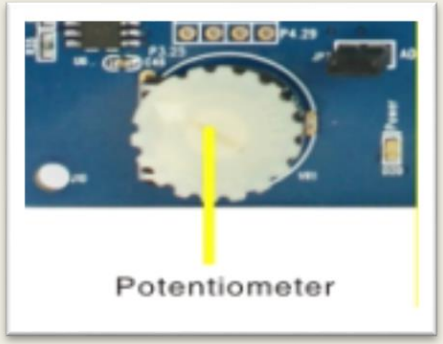
SSP1 (Synchronous Serial Port 1) [Driver_SPI1] ☑

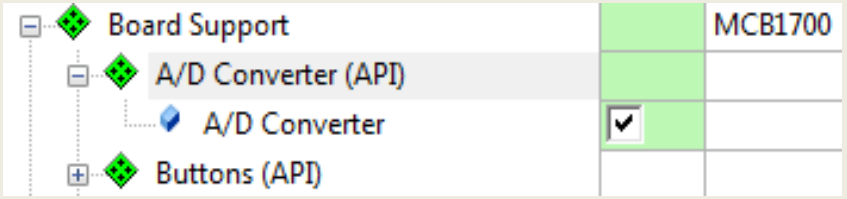
Pin Configuration

SSP1_SSEL	P0_6
SSP1_SCK	P0_7
SSP1_MISO	P0_8
SSP1_MOSI	P0_9

8. ADC

- **Reading the signal from potentiometer**

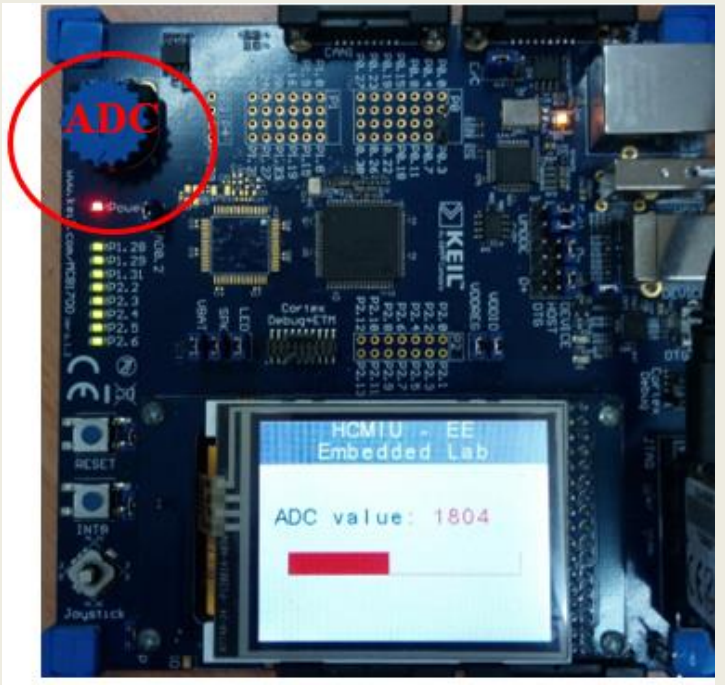




- Basing library **ADC_MBC1700.c**

main.c | ADC_MCB1700.c

```
1  /*----------------------------------------------------------------------
2   * Name:    ADC_MCB1700.c
3   * Purpose: A/D Converter interface for MCB1700 evaluation board
4   * Rev.:    1.00
5   *----------------------------------------------------------------------
6
7  /* Copyright (c) 2013 - 2014 ARM LIMITED
```



The result

Thanks for your reading

- Please, linking to the Instruction section to do step by step on MDK. (The project folder was attached)