Embedded Systems Laboratory

School of Computer Science and Electronic Engineering

University of Essex

ARM Main Board Hardware Description

Overview

The embedded systems laboratory uses one of the Atmel AT91 ARM Thumb-based microcontrollers. A flexible hardware configuration is provided consisting of a standalone main microcontroller board with provision to accept a number of different daughter boards.

The ARM Main Board

The ARM main board (Figure 1) is designed to be a standalone unit providing the basic functionality required for the development of an embedded system. The board consists of an Atmel AT91SAM7S256 microcontroller with 256Kbytes flash memory and 64Kbytes sram memory, eight push button switches, eight tri-colour LEDs, a 160x128 OLED colour display, usb port and JTAG programming port. In addition the main board also provides two expansion ports to allow daughter boards to be used.

See datasheets for a more detailed explanation of the microcontroller and OLED display.

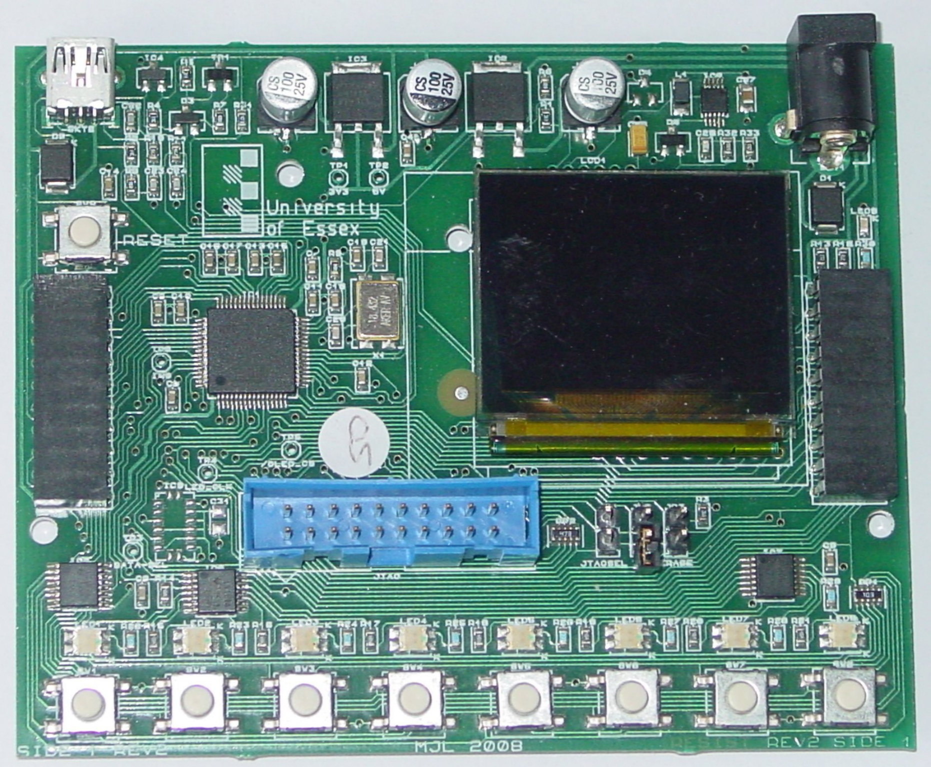


Figure 1: ARM main board

Configuration

Initialisation code provided within the development environment provides the following default configuration for the processor and the main board.

Processor:

SRAM:	0x00200000 – 0x0020FFFF
(Mirrored):	0x00000000 – 0x0000FFFF
FLASH:	0x00100000 – 0x0013FFFF
Master Clock:	48Mhz
Watchdog timer:	Disabled
User Reset:	Enabled

Main Board: (PIO configuration)

PA0	Peripheral A - PWM0
PA1	PIO - Output
PA2	Peripheral A - PWM2
PA3	Peripheral A - TWD (I^{2}C Data)
PA4	Peripheral A - TWCK (I^{2}C Clock)
PA5	Peripheral A - RXD0 (Serial Port 0)
PA6	Peripheral A - TXD0 (Serial Port 0)
PA7	PIO - Output
PA8	PIO - Output
PA9	Peripheral B - NPCS1 (SPI Device 1)
PA10	Peripheral B - NPCS2 (SPI Device 2 - OLED Display)
PA11	Peripheral A - NPCS0 (SPI Device 0)
PA12	Peripheral A - MISO (SPI In) Pull up
PA13	Peripheral A - MOSI (SPI Out)
PA14	Peripheral A - SPCK (SPI Clock)
PA15	Peripheral A - TF (I^{2}S)
PA16	Peripheral A - TK (I^{2}S)
PA17	Peripheral A - TD (I^{2}S)

PA18	Peripheral A - RD (I^{2}S)
PA19	PIO - Input
PA20	Peripheral A - RF (I^{2}S)
PA21	Peripheral A - RXD1 (Serial Port 1)
PA22	Peripheral A - TXD1 (Serial Port 1)
PA23	PIO - Input
PA24	PIO - Output (LED Clock)
PA25	PIO - Output (Button Select, OLED Command/Data Select, LED Data)
PA26	PIO - Input (Button Input)
PA27	PIO - Input (Button Input)
PA28	PIO - Input (Button Input)
PA29	PIO - Input (Button Input)
PA30	Peripheral A - IRQ1
PA31	Peripheral B - PCK2

Each SPI device can be configured separately, at initialisation all devices are configured as follows: Master Mode, Protocol Mode 0, 8-bit data transfer, 12Mhz clock rate, 3ns delay between consecutive transfers. See datasheet for more details.

Serial port 0 is initialized as follows: 9600 baud, 8-bit data transfer, 1 stop bit, no parity. See datasheet for more details.

All other devices are not initialized.

Expansion Ports

The main board provides two expansion ports; the signal configuration of the ports is shown below. The connections are shown looking directly at the interface ports. The ports are reversed so that daughter boards can be connected to either port without sustaining electrical damage. Not that not all signals are common to both interface ports. See AT91 microcontroller datasheet for details of the signals.

Left Expansion Port (SKT5)

Gnd	20	19	AD7
AD6	18	17	PCK2
PA19	16	15	PA23
TK	14	13	RF
RD	12	11	TF
TD	10	9	SPI Device Select 0
/Reset	8	7	TWCK
TWD	6	5	SPCK
MISO	4	3	MOSI
5v	2	1	3v3

Right Expansion Port (SKT6)

3v3	1	2	5v
MOSI	3	4	MISO
SPCK	5	6	TWD
TWCK	7	8	/Reset
SPI Device Select 1	9	10	Serial Port 0 Tx
Serial Port 1 Tx	11	12	Serial Port 0 Rx
Serial Port 1 Rx	13	14	IRQ1
PWM0	15	16	PWM2
PA7	17	18	PA8
Gnd	19	20	Gnd

JTAG Port

The JTAG programming port is located on the top side of the main board just above the row of LEDs, see Figure 1 (the JTAG port is the blue socket). The JTAG port uses the boundary scan features of the processor to provide direct access and control to various processor features such as memory and internal registers, but also provides external control of the processor. Using the JTAG port programs can be loaded, executed and debugged on the processor.

Access to the JTAG functionality is via a JTAG probe (Figure 3), the probe connects to the host computer and allows remote programs to access the processors functionality.

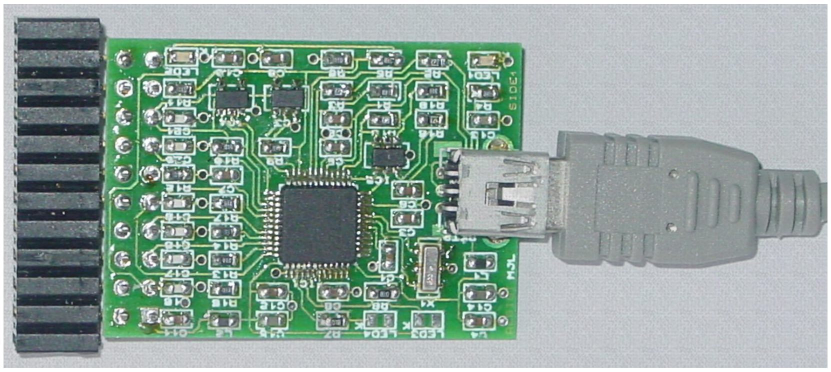


Figure 2: JTAG probe

The JTAG probe should be connected to the main board as shown below (Figure 3), components away from the LEDs. The connectors are not keyed, so it is possible to insert the probe incorrectly. If this is done, no damage will occur, but there will be no functionality.

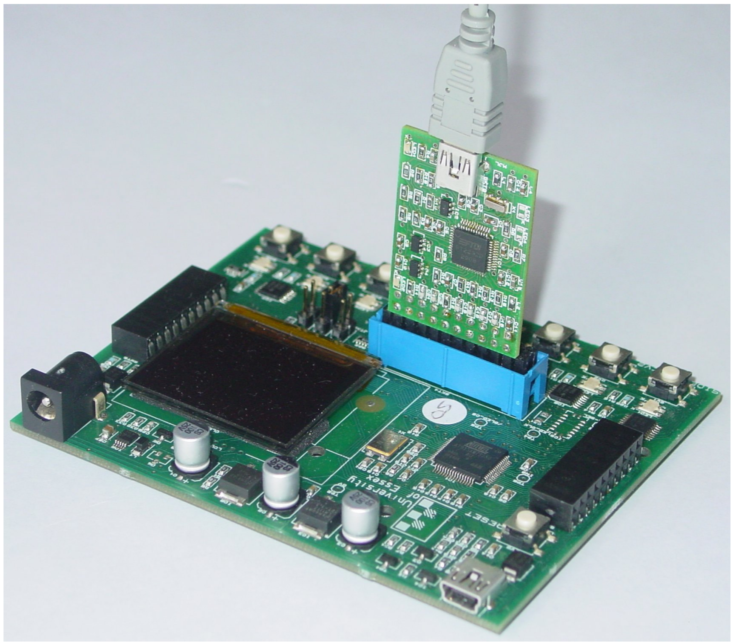


Figure 3: JTAG probe inserted into main board

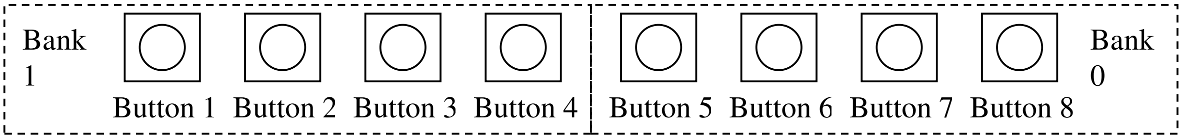
Main Board Devices

The Push Button Switches

Description

Eight push to make switches are provided along the bottom edge of the ARM main board. Each switch provides a binary input via the microcontroller parallel interface. In each case a zero represents the switch pressed and a one represents the switch released. The switches are arranged as two banks of four switches plus a bank select output, requiring only five parallel connections for the eight switches.

The silk screen legend on the circuit board labels the buttons as SW1 to SW8 from left to right. The right hand bank containing buttons SW5 to SW8 is bank 0 (select line low). The right hand bank containing buttons SW1 to SW4 is bank 1 (select line high). See figure below.



PIO Pins Used

I/O Line	Peripheral	Direction	Function
PA25	A	Output	Select button bank, Bank 0 = 0, Bank 1 = 1 Warning: PA25 is a multifunction pin and is used elsewhere
PA26	A	Input	Read input from Button 4 (select = 1) or Button 8 (select = 0)
PA27	A	Input	Read input from Button 3 (select = 1) or Button 7 (select = 0)
PA28	A	Input	Read input from Button 2 (select = 1) or Button 6 (select = 0)
PA29	A	Input	Read input from Button 1 (select = 1) or Button 5 (select = 0)

Usage

To read the value of a button the select bank output (PA25) must first be written to select the required bank. The value of the four buttons contained the selected bank can now be read. The banks can be read in any order.

Example

```
/*
 * Determine value of Button 3
 */

AT91C_BASE_PIOA->PIO_SODR = AT91C_PIO_PA25;
```

```
if ((AT91C_BASE_PIOA->PIO_PDSR & AT91C_PIO_PA27) == 0)
    /* Button is pressed */
else
    /* Button is released */
```

The Tri-colour LEDs

Description

Eight tri-colour LEDs are provided above the push button switches on the ARM main board. Each LED can either be off, red, green or orange (both red and green). The silk screen legend on the circuit board labels the LEDs as LED1 to LED8 from left to right.

The LEDs are arranged as sixteen bit shift register, requiring only two parallel connections for the eight LEDs. Each LED represents 2 bits of the shift register, the high bit displays the colour red and the low bit displays the colour green. (Setting both bits displays an orange colour.)

PIO Pins Used

I/O Line	Peripheral	Direction	Function
PA24	A	Output	LED clock pin.
PA25	A	Output	LED data pin Warning: PA25 is a multifunction pin and is used elsewhere

Usage

When the clock pin (PA24) is taken high the LED data register is shifted one position left. The high order data bit is lost and the low order data bit becomes the current value of the data pin (PA25). All intermediate values move one position to the left.

The value of a single LED can not be changed without altering the values of all the other LEDs. Therefore any program should keep an internal representation of the current state of the LEDs and update this representation with any changes before writing the complete representation to the LEDs. Thus it is always necessary to perform 16 shift operations to alter the value of any single or any combination of LEDs.

Example

```
/*
 * Internal representation of LEDs.
 * Each value can be OFF(0), RED(2), GREEN(1) or ORANGE(3)
 *
 * Value of LEDstate is set elsewhere.
 */

short LEDstate[8];

/*
 * For each LED apply two clocks.
 * One for the ‘red’ bit or one for the ‘green’ bit
 */
for (i = 0; i < 8; i++)
{
    /*
```

```
         * Set Data bit hi or lo
         */
        if (LEDstate[i] == RED || LEDstate[i] == ORANGE)
            AT91C_BASE_PIOA->PIO_SODR = AT91C_PIO_PA25;
        else
            AT91C_BASE_PIOA->PIO_CODR = AT91C_PIO_PA25;

        /*
         * Then apply clock
         */
        AT91C_BASE_PIOA->PIO_SODR = AT91C_PIO_PA24;
        AT91C_BASE_PIOA->PIO_CODR = AT91C_PIO_PA24;

        /*
         * Set Data bit hi or lo
         */
        if (LEDstate[i] == GREEN || LEDstate[i] == ORANGE)
            AT91C_BASE_PIOA->PIO_SODR = AT91C_PIO_PA25;
        else
            AT91C_BASE_PIOA->PIO_CODR = AT91C_PIO_PA25;

        /*
         * Then apply clock
         */
        AT91C_BASE_PIOA->PIO_SODR = AT91C_PIO_PA24;
        AT91C_BASE_PIOA->PIO_CODR = AT91C_PIO_PA24;
    }
```

The OLED Colour Display

Description

A 160x128 dot colour OLED display is provided on the ARM main board. The display supports only pixel level write operations. See datasheet for more details.

Communication with the display is via the SPI bus. The display only supports 8-bit SPI write operations, read operations are not supported.

PIO Pins Used

I/O Line	Peripheral	Direction	Function
PA10	B	Output	OLED Display Select (SPI Device Select 2)
PA13	A	Output	Data Out (SPI MOSI)
PA14	A	Output	Clock (SPI Clock)
PA25	A	Output	Command/Data Select Warning: PA25 is a multifunction pin and is used elsewhere

Usage

Messages to the display take the form <Command><Parameters>. To write a command the Command/Data Select line (PA25) must be taken LO before transmitting the command via the SPI bus. It should then be taken HI before any parameters are sent.

The OLED display is connected the SPI device select 2, this output must be asserted to communicate with the display.