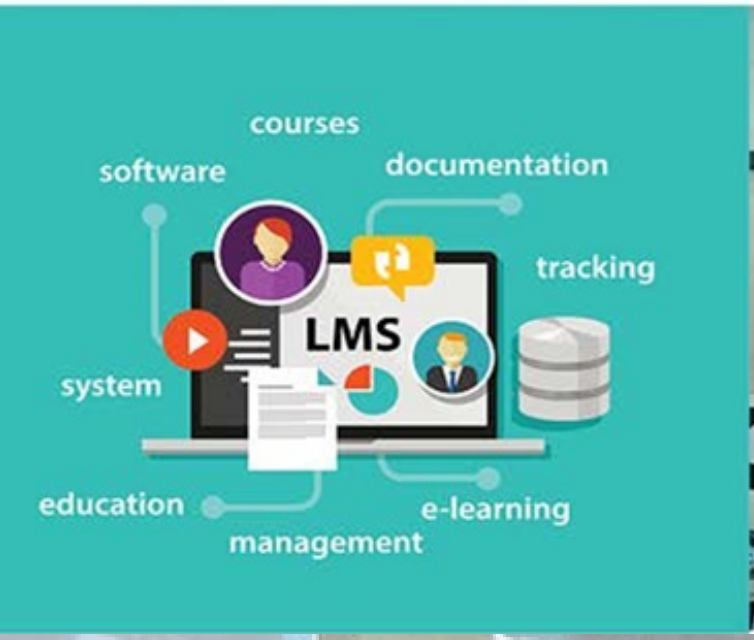
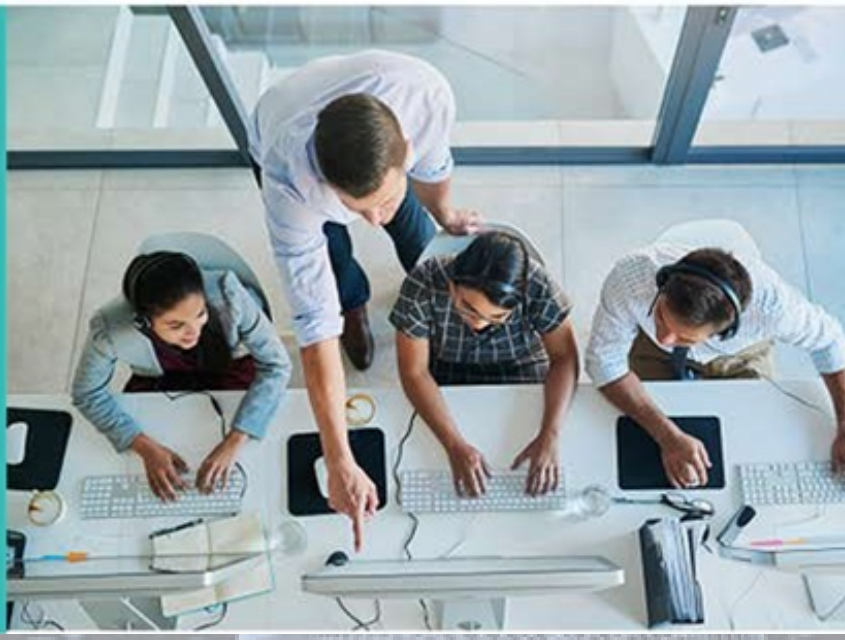
Skill AP
APSSDC

Andhra Pradesh State Skill Development Corporation





Embedded systems

Introduction to ARM7 (LPC2148)

ARM7

INTRRODUCTION:

An embedded system combines mechanical, electrical, and chemical components along with a computer, hidden inside, to perform a single dedicated purpose. There are more computers on this planet than there are people, and most of these computers are single-chip microcontrollers that are the brains of an embedded system. Embedded systems are a ubiquitous component of our everyday lives. We interact with hundreds of tiny computers every day that are embedded into our houses, our cars, our bridges, our toys, and our work. As our world has become more complex, so have the capabilities of the microcontrollers embedded into our devices. Therefore, the world needs a trained workforce to develop and manage products based on embedded microcontrollers.

About ARM7:

The full form of an ARM is an advanced reduced instruction set computer machine, and it is a 32-bit processor architecture expanded by ARM holdings. The applications of an ARM processor include several microcontrollers as well as processors. The architecture of an ARM processor was licensed by many corporations for designing ARM processor-based SoC products and CPUs. This allows corporations to manufacture their products using ARM architecture. Likewise, all main semiconductor companies will make ARM-based SOCs such as Samsung, Atmel, etc.

ARM7 processor is commonly used in embedded system applications. Also, it is a balance among classic as well as a new-Cortex sequence. This processor is tremendous in finding the resources existing on the internet with excellent documentation offered by NXP Semiconductors. It suits completely for an apprentice to obtain in detail hardware & software design implementation. The LPC2148 microcontroller is designed by Philips with several in-built features & peripherals. Due to these reasons, it will make it more reliable as well as an efficient option for an application developer. LPC2148 (ARM7) is a 16-bit or 32-bit microcontroller based on the ARM7 family.

Features of LPC2148:

LPC2148 microcontroller has developed by Philips (NPX semiconductor) company It has so many built-in peripherals and features. Therefore, it has become so efficient and reliable as compared to other microcontrollers. It is a 16 bit or 32-bit ARM 7 based microcontroller that has different features.

- ✓ It is 16- or 32-bit ARM 7 family-based microcontrollers and is available in market in small packages such as LQFP64.
- ✓ Its programming time is 1 millisecond for 256 bytes and 400 milliseconds for erasing the on full chip data.

- ✓ It is used on-chip bootloader software during the ISP (in system programming) and IAP (in application programming)

- ✓ It has 8 kB to 40 KB on chip static RAM and 32 kB to 512 kB on chip flash memory.
- ✓ It offers the high-speed operation at frequency 60 MHz with wide range of interface almost 128 bits.

- ✓ LP218 has clock input with 32 K Hz frequency and low power RTC (real time clock).
- ✓ It has embedded ICE RT and embedded trace interface which offers the tracing of instruction execution with high speed with real time debugging.

- ✓ It offers a changeable output with 10-bit DAC (digital to analog converter).
- ✓ LPC2148 offers the changeable input with 10-bit ADC (analog to digital converter) with very conversion time such as 2.44µs / channel.
- ✓ For counting the external events it has two 32-bit timers, watchdog timer and PWM unit.

- ✓ It has several serial interfaces such as two 12C buses, two 16C550 UARTs with 400 Kbit speed.

- ✓ The modes, which are used for power conversion are called idle and power down.
- ✓ In this LQFP64 small package, it also has 5 volts input output pins for any general purposes.

- ✓ It has 1 MHz to 25 MHz on chip incorporated oscillator which works as an exterior crystal.
- ✓ It also has individual enable or disable peripheral function and peripheral SLK scaling for extra power optimization.

LPC2148 Pin Configuration

GPIO (General-purpose input/output) is a pin on an Integrated Circuit. It can be either an input pin or output pin, whose behaviour can be controlled at the run time. A group of these pins is called a port (for example, Port 0 of LPC2148 has 32 pins).
The LPC2148 has two 32-bit General Purpose I/O ports.

1. **PORT0**
2. **PORT1**

PORT0 is a 32-bit port

- ❖ Out of these 32 pins, 28 pins are configured as either general-purpose input or output.
- ❖ 1 of these 32 pins (P0.31) can be configured as general-purpose output only.
- ❖ 3 of these 32 pins (P0.24, P0.26, and P0.27) are reserved. Hence, they are not available for use.

PORT1 is also a 32-bit port.

- Only 16 of these 32 pins (P1.16 – P1.31) are available for use as general-purpose input or output.

Note: The Port 0 pins do not have a built-in pull-up or pull-down resistors. Hence, while using GPIOs on Port 0, in some cases, we need to connect pull-up or pull-down resistors externally.

Almost every pin of these two ports has some alternate function available. For example, P0.0 can be configured as the TXD pin for UART0 or as PWM1 pin as well. The functionality of each pin can be selected using the **Pin Function Select Registers.**

Pin Function Select Registers

Pin Function Select Registers are 32-bit registers. These registers are used to select or configure specific pin functionality.

There are 3 Pin Function Select Registers in LPC2148:

1. **PINSEL0: -** PINSEL0 is used to configure PORT0 pins P0.0 to P0.15.
2. **PINSEL1: -** PINSEL1 is used to configure PORT0 pins P0.16 to P0.31.
3. **PINSEL2: -** PINSEL2 is used to, PORT1 pins P1.16 to P1.31.

These configuration registers are of 32-bit wide registers. Any pin on the LPC2148 can have a maximum of 4 functions. Hence in order to select one of the four functions, two corresponding bits of the PINSEL register are needed. So, a 32-bit PINSEL register can control 16 pins with 2-bits to control each pin.

Example:

PINSEL0	Pin name	Function of 00	Function of 01	Function of 10	Function of 11	Rest value
1:0	P0.0	GPIO	TXD(UART0)	PWM1	Reserved	00

GPIO Registers

GPIO registers that control the GPIO operations. There are 4 GPIO Registers available to control PORT0 and PORT1.

1. **IODIRx (GPIO Port Direction control register):** This is a 32-bit wide register. This register individually controls the direction of each port pin. Setting a bit too logic 1 configures the corresponding pin as an output pin. Setting a bit too logic 0 configures the corresponding pin as an input pin.

Example:

```
IODIR0 = (1<<5);      // pin P0.5 set to 1 and other pins set to 0 and other pins affected

IO0DIR |= 0x0000002;          // or 0x2; (hexadecimal - OR and assign: other pins not affected)

IO0DIR |= (1<<2);     // (binary - OR and assign: other pins not affected)
```

2. **IOSETx (GPIO Port Output Set register):** This is a 32-bit wide register. This register is used to make pins of Port (PORT0/PORT1) HIGH. Writing one to specific bit makes that pin HIGH. Writing zero does not affect.

Example1:

Consider that we want to configure Pin 20 of Port 0 i.e. P0.20 as Output and want to drive it High (Logic 1). This can be done as:

```
IODIR0 |= (1<<20);      // Config P0.20 as Output
IOSET0 |= (1<<20);     // Make output High for P0.20 (logic 1 assign to P0.20).
```

Example2:

Configuring P0.15 and P0.20 as Output and Setting them High:

```
IODIR0 |= (1<<15) | (1<<20); // Config P0.15 and P0.20 as Output
IOSET0 |= (1<<15) | (1<<20); // Make output High for P0.15 and P0.20
```

Example3:

Configuring 1st 16 Pins of Port 0 (P0.0 to P0.15) as Output and Setting them High:

```
IO0DIR |= 0xFFFF; // Config P0.0 to P0.15 as Output
IO0SET |= 0xFFFF; // Make output High for P0.0 to P0.15
```

3. **IOCLR0 (GPIO Port Output Clear register):** This is a 32-bit wide register. This register is used to make pins of Port LOW. Writing one to specific bit makes that pin LOW. Writing ‘0’ does not affect.

Example:

a) Configure pin P0.0 as output pins
```
IO0DIR = 0x00000001
```

b) Configure pin P0.0 as an output. Then set that pin HIGH.
```
IO0DIR = 0x00000001
IO0SET = (1<<0);
```

c) Configure pin P0.0 as an output. Then set that pin LOW.

IO0DIR = 0x00000001; OR IO0DIR = (1<<0);
IO0CLR = (1<<0);

Note: - When IOSET = 1 it will set the particular pin, but it doesn't mean that to clear the pin you will use IOSET = 0 or IOCLR = 0, this is different in case of ARM, to set the pin we have to use IOSET=1 and to clear we have to use IOCLR=1.

4. **IOPIN0 (GPIO Port Pin value register):** This is a 32-bit wide register. This register is used to read/write the value on Port (PORT0/PORT1). But care should be taken while writing. Masking should be used to ensure write to the desired pin. IOPIN Register is used to get the status of the pins.

Example:

a) Writing 1 to P0.4 using IO0PIN
IO0PIN = IO0PIN | (1<<4)

b) Writing 0 to P0.4 using IO0PIN
IO0PIN = IO0PIN & (~ (1<<4))

c) Writing F to P0.7-P0.4
IO0PIN = IO0PIN | (0x000000F0)