1

Introduction to Arm® Mbed™

Isn’t it a pleasure to study and practice what you have learned?

- Confucius

1.1 What is an Embedded System?

An embedded system is a small-scale computer system that is part of a machine or a larger electrical/mechanical system. It is often designed to perform certain dedicated tasks and often a real-time system. It is called embedded because the computer system is embedded within a hardware device. Embedded systems are important, as they are getting increasingly used in many daily appliances, such as digital watches, cameras, microwave ovens, washing machines, boilers, fridges, smart TVs, and cars. Embedded systems also often need to be small in size, low in cost, and have low power consumption.

Figure 1.1 shows the schematic diagram of a typical embedded system that includes a microcontroller, inputs/outputs, and communication interfaces.

Microcontroller

Microcontroller is the brain of an embedded system, which orchestrates all the operations. A microcontroller is a computer processor with memory and all input/output peripherals on it. More details about microcontrollers will be illustrated in the next section.

Inputs

An embedded system interacts with the outside world through its inputs and outputs. Inputs can be digital inputs or analog inputs. Inputs are typically used for reading data from sensors (temperature sensor, light sensor, ultrasound sensor, etc.) or other types of input devices (keys, buttons, etc.).

Outputs

Outputs can also be digital outputs or analog outputs. Outputs are typically used for display, driving motors, or other devices (actuators).
An embedded system communicates with other devices using communication interfaces, which includes Ethernet, USB (Universal Serial Bus), CAN (Controller Area Network), Infrared, ZigBee, WiFi and Bluetooth, for example.

1.2 Microcontrollers and Microprocessors

At the heart of embedded systems are microcontrollers. Although there are embedded systems built on microprocessors, modern embedded systems are largely based on microcontrollers. A typical microcontroller contains a central processing unit (CPU), interrupts, timer/counter, memory, and other peripherals, all in a single integrated circuit (IC). A microcontroller is a true computer on a chip or system-on-a-chip (SoC). Microcontrollers are ideal for control applications because you can use them to build an embedded system with little additional circuitry.

Microcontrollers (MCU or μC) are different from microprocessors (MPU). A microprocessor is a single IC with only a central processing unit (CPU) on it. In order to make it functional, you will need to add external memory and other peripheral devices. Figure 1.2 shows the main differences between a microprocessor and a microcontroller. To put it simply, you can imagine that a microprocessor is just a CPU on a single IC, while a microcontroller is a small computer with CPU, memory, and other peripherals.

Microprocessors are mainly used in general-purpose systems like personal computers. They have relatively high computational capacity and can perform numerous tasks. Microprocessors have relatively high clock frequency, usually in the order of gigahertz. Microprocessors generally consume more power and often require external cooling system.

Microcontrollers are designed for control applications and are generally used in embedded systems. They have relatively low computational capacity and can perform single or very few tasks. Microcontrollers have relatively low clock frequency, usually in the order of megahertz. Microcontrollers consume less power and have no need for a cooling system.

Figure 1.3 shows a more detailed schematic diagram of a microcontroller. Following are its key components.
CPU

CPU, often referred to as a processor or central processor, is the brain of a microcontroller. See Figure 1.4 for details. It contains three main components: the arithmetic logic unit (ALU), the control unit, and registers. ALU performs arithmetic and logical operations, registers provide operands to the ALU and store the results of ALU operations, and the control unit controls the overall operations and communicates with both ALU and registers. The operation cycle of CPU can be described as fetch, decode, and execute.

CPU communicates with its external peripherals such as memory and input/output through system bus, which includes a data bus, an address bus, and a control bus. The data bus is for carrying information, the address bus is for determining where the information should be sent, and the control bus is for determining the operation. The address

Figure 1.2 Comparison between a microprocessor and a microcontroller.

Figure 1.3 Detailed schematic diagram of a microcontroller.
bus is unidirectional, from CPU to peripherals, while the data bus and control bus are bidirectional.

CPU can be divided into different types, depending on the instruction set implemented. The instruction set, also called instruction set architecture (ISA), is a set of basic operations that CPU can perform. Two main types are complex instruction set computing (CISC) and reduced instruction set computing (RISC). A CISC CPU has very large instruction sets (300 and more) and more complex hardware, but has more compact software code. It takes more cycles per instruction. It also uses less RAM, as there is no need to store intermediate results. A RISC CPU has small instruction sets (100 and less) and simpler hardware, but has more complicated software code. It takes one cycle per instruction and uses more RAM to handle intermediate results. Typical examples of CISC CPUs are AMD and Intel x86, which are mainly used in personal computers, workstations, and servers, as they are capable of more sophisticated tasks. Typical examples of RISC CPUs are Atmel AVR, PIC, and ARM®, which are mainly used in microcontrollers because they consume less power.

Memory
Microcontrollers use memories for storing programs and data. There two types of memory, internal and external. Internal memory is limited in size but is fast. For applications in which internal memory is not enough, external memory is then needed. Traditionally, there are two types of external memory, random access memory (RAM) and read-only memory (ROM). RAM can be accessed randomly and you can perform both read and write operations to RAM. RAM will lose all its contents when power is switched off. ROM is read-only memory, which means you can read data from it but cannot write data to it. ROM does not lose its contents even if power is switched off; therefore, it is used to store programs and data permanently.
However, there are new types of memories, such as electrically erasable programmable ROM (EEPROM) and non-volatile RAM (NVRAM). Both can be read and write and do not lose its contents even if power is switched off. Flash memory is the best example of NVRAM. It is high-density, low-cost, fast, and electrically programmable. Flash memory is being extensively used for embedded systems that contain embedded operating systems and the application program.

Parallel Input/Output Ports
Parallel input/output ports have multiple wires (or pins) running parallel to each other. It is called parallel because multiple signals can be accessed all at once. Parallel input/output ports are mainly used to drive/interface various devices such as LCDs, LEDs, printers, memories, and so on to a microcontroller. Parallel ports can transfer data much faster than serial ports, but only suitable for short distance communications due to interference and noise.

Serial Input/Output Ports
Serial input/output ports use a single data wire to transfer data. Serial ports therefore are much slower than parallel input/output ports. However, serial ports can have higher bandwidth, and can be used over longer distances. Universal Asynchronous Receiver/Transmitter (UART) peripheral is a commonly used serial input/output port in embedded systems. It uses one wire for receiving data (Rx) and one wire for transmitting data (Tx).

Timers/Counters
Timers and counters are useful functions for a microcontroller. A microcontroller may have more than one timer and counters. The timers and counters provide all timing and counting functions inside the microcontroller, including clock functions, modulations, pulse generations, frequency measuring, and making oscillations.

Analog to Digital Converter (ADC)
ADC converts analog signals to digital signals. It is mainly used for reading voltage outputs of sensors. ADC can be 8 bits, 10 bits, 12 bits, 16 bits, 24 bits, and even 32 bits. The higher the number of bits means the higher conversion resolution. The bandwidth of an ADC (i.e., the range of frequencies it can measure) is determined by its sampling rate or sampling frequency. According to Nyquist–Shannon sampling theorem, the highest frequency that an ADC can measure is less than half of its sampling rate. The typical ADC sampling rate of mbed boards is about a few hundreds kilohertz.

Digital to Analog Converter (DAC)
DAC is the opposite of ADC. DAC converts the digital signals into analog signals. It usually used for controlling analog devices such as audio speakers, DC motors, and various drives.

Interrupt Control
Interrupt is one of the most important and powerful features in microcontroller applications. The interrupt control is used to interrupt a working program. The interrupts
can be either hardware interrupts (external, activated by using interrupt pin) or software interrupts (internal, by using interrupt instruction during programming).

**Reset**
Reset is an important function that exists in all microcontrollers. Reset can make sure microcontrollers go back to its original state. This is important, especially when things go wrong.

**Watchdog**
A watchdog, or watchdog timer, is a piece of electronic hardware that is commonly used in embedded systems to automatically detect software malfunctions and to reset the processor. A watchdog timer basically counts down from some initial value to zero. The embedded software selects the counter's initial value and periodically restarts it. If the counter ever reaches zero before the software restarts it, the software is presumed to be malfunctioning and the processor will be reset.

### 1.3 ARM® Processor Architecture

ARM® (Advanced RISC Machine) architecture is a computer processor architecture based on reduced instruction set computing (RISC). ARM® architecture was originally developed by British company Acorn Computers based in Cambridge, United Kingdom in the 1980s. ARM® originally stood for Arcon RISC Machine. The first ARM processors were used in BBC Microcomputers. Acorn started working with Apple Computer and VLSI Technology in the late 1980s. In 1990, Acorn spun off the design team into a new company named Advanced RISC Machines (ARM®) Ltd. The company name was later changed to ARM® Holdings plc. ARM® Holdings plc floated on the London Stock Exchange and NASDAQ in 1998. It became a member of the FTSE 100 in 1999.

ARM® processors become increasingly popular after being used on Apple's iPhone and iPad since 2007. To date, ARM® processors are widely used in smartphones, tablets, and smart TVs. Over 50 billion ARM® processors were produced as of 2014. In July 2016, ARM® Holdings has an annual turnover about £1 billion and agreed to a £24.3 billion takeover by Japan's Softbank company. The takeover is largely seen as an investment for the Internet of Things (IoT), in which ARM® processors will be likely taking a dominant role.

To date, ARM® processors can be generally divided into three categories: Application, Real-time, and Microcontroller, as shown in Table 1.1. The ARM® application processors (Cortex-A series) are the most powerful processors, optimized for higher performance, and can be typically used in phones, pads, tablets, and computers. The ARM® Real-time processors (Cortex-R series), optimized for faster response, can be typically used in industrial, home, and automotive applications. The ARM® Microcontroller processors (Cortex-M series), optimized for smaller size, and lower power consumption, can be typically used in embedded systems, and, of course, IoT applications!

Figure 1.5 shows the performance functionality and capacity of the ARM® Cortex-A, Cortex-R, and Cortex-M processors.
Table 1.2 shows the different Cortex-M microcontrollers. The Cortex-M0, Cortex-M0+, and Cortex-M23 controllers are designed for the lowest power consumptions. The Cortex-M3, Cortex-M4, and Cortex-M33 controllers are designed for the highest efficiency. The Cortex-M7 controllers are designed for the highest performance. In this book, we will focus only on the ARM® Microcontroller processors, specifically the Cortex-M4 series.

Figure 1.6 shows the features and functions of ARM® Cortex-M series processors.

Further Information about ARM® Processor Architecture

https://en.wikipedia.org/wiki/ARM_architecture
The Arm® Mbed™ is a platform and operating system based on 32-bit ARM® Cortex-M microcontrollers. It is collaboratively developed by ARM® and its technical partners, and is designed for Internet of Things (IoT) devices. It provides the operating system, cloud services, tools, and developer ecosystem to make the creation and deployment of IoT solutions possible.

One of the major features of the Arm® Mbed™ systems is its web-based development environment. Just plug the device into computer using a USB cable, which will appear on your computer as a USB memory stick. Write and compile your software code using the Arm® Mbed™ Online Compiler, download the compiled code into the device, and press the onboard reset button to run!

Table 1.2 The Cortex-M Series Microcontrollers

<table>
<thead>
<tr>
<th>Lowest Power and Area</th>
<th>Performance Efficiency</th>
<th>Highest Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cortex-M23</td>
<td>TrustZone in smallest area, lowest power</td>
<td>Cortex-M33</td>
</tr>
<tr>
<td>Cortex-M0+</td>
<td>Highest energy efficiency</td>
<td>Cortex-M4</td>
</tr>
<tr>
<td>Cortex-M0</td>
<td>Lowest cost, low power</td>
<td>Cortex-M3</td>
</tr>
<tr>
<td></td>
<td>Freely available for design and simulation via DesignStart</td>
<td>Cortex-M7</td>
</tr>
</tbody>
</table>

Figure 1.6 The features and functions of ARM® Cortex-M series processors from the ARM® website.
(Source: https://community.arm.com/processors/b/blog/posts/meet-the-new-arm-cortex-m7-processor-supercharging-embedded-devices)
Arm® Mbed™ provides all you need to develop IoT and embedded devices. It has a full support for over 100 mbed-enabled boards and more than 400 components. It also has tools for writing, building, and testing applications, and server and client-side tools to communicate with your devices.

The mbed microcontrollers provide experienced embedded developers a powerful and productive platform for building proof-of-concepts. For developers new to 32-bit microcontrollers, mbed provides an accessible prototyping solution to get projects built with the backing of libraries, resources, and support shared in the mbed community.

Figure 1.7 and Figure 1.8 show the Arm® Mbed™ home page, and the corresponding developer website. Figure 1.9 shows a list of development boards that is supported by the Arm® Mbed™. There are several development boards worth mentioning:

### 1.4.1 NXP LPC1768

This is one of the most popular development boards. It is based on the NXP LPC1768 microcontroller, with a 32-bit ARM® Cortex-M3 core running at 96 MHz. It has 512 KB
flash, 32 KB RAM and lots of interfaces, including built-in Ethernet, USB host and device, CAN, SPI, I2C, ADC, DAC, PWM, and other I/O interfaces. The 12 bits of ADC are particularly useful. Figure 1.10 shows the board and its pinouts, including commonly used interfaces and their locations. The pins P5-P30 can also be used as DigitalIn and DigitalOut interfaces.

Features

- NXP LPC1768 MCU
  - High-performance ARM® Cortex™-M3 Core
  - 96 MHz, 32 KB RAM, 512 KB flash
  - Ethernet, USB host/device, 2×SPI, 2×I2C, 3×UART, CAN, 6×PWM, 6×ADC (12 bits), GPIO
Introduction to Arm® Mbed™

Prototyping form-factor
- 40-pin 0.1" pitch DIP package, 54×26mm
- 5V USB or 4.5-9V supply
- Built-in USB drag ‘n’ drop flash programmer

mbed.org developer website
- Lightweight online compiler
- High level C/C++ SDK
- Cookbook of published libraries and projects

There is also an mbed Application Board for NXP LPC1768 (Figure 1.11. The NXP LPC1768 and its mbed application board make a great learning kit.

Features
- 128 × 32 graphics LCD
- 5 way joystick
- 2 × potentiometers
- 3.5 mm audio jack (analog out)
- Speaker, PWM connected
- 3 Axis +/-1.5g accelerometer
- 3.5mm audio jack (analog in)
- 2 × Servo motor headers
- RGB LED, PWM connected
- USB-mini-B connector
- Temperature sensor

Figure 1.10 The NXP LPC1768 development board and its pinout from the Arm® Mbed™ website. (Source: https://os.mbed.com/platforms/mbed-LPC1768/)
Notes

The new application board has been designed to enable the maximum number of potential experiments and projects, with the minimum footprint.

Figure 1.11 The mbed application board for NXP LPC1768 development board, front (left) and back (right), from the Arm® Mbed™ website. (Source: https://os.mbed.com/components/mbed-Application-Board/)

- Socket for Xbee (Zigbee) or RN-XV (WiFi)
- RJ45 Ethernet connector
- USB-A connector
- 1.3 mm DC jack input

Further Information about LPC1768

https://os.mbed.com/platforms/mbed-LPC1768/
https://os.mbed.com/components/mbed-Application-Board/

1.4.2 NXP LPC11U24

This is another interesting development board. It is based on the NXP LPC11U24, with a 32-bit ARM® Cortex-M0 core running at 48 MHz. It includes 32 KB flash, 8 KB RAM, and lots of interfaces, including USB device, SPI, I2C, ADC, and other I/O interfaces. Figure 1.12 shows the board and its printout, including the commonly used interfaces and their locations. The pins P5–P30 can also be used as DigitalIn and DigitalOut interfaces.

Different from NXP LPC1768, NXP LPC11U24 is much slower and less powerful, but it uses less power and is much cheaper, so it is mainly designed for low-cost USB devices and battery-powered applications.

Features

- NXP LPC11U24 MCU
- Low power ARM® Cortex™-M0 core
Introduction to Arm® Mbed™

- 48 MHz, 8 KB RAM, 32 KB flash
- USB device, 2×SPI, I2C, UART, 6×ADC, GPIO
- Prototyping form-factor
- 40-pin 0.1” pitch DIP package, 54×26mm
- 5V USB, 4.5–9V supply or 2.4–3.3V battery
- Built-in USB drag ’n’ drop flash programmer
- mbed.org developer website
- Lightweight online compiler
- High-level C/C++ SDK
- Cookbook of published libraries and projects

Further Information about LPC11U24

https://os.mbed.com/platforms/mbed-LPC11U24/

1.4.3 BBC Micro:bit

The BBC micro:bit is a pocket-sized, codable computer, developed by BBC through a major partnership with 31 organizations, including ARM®, NXP, element14, Microsoft,
and Cisco. Figure 1.13 shows the board and its printout. It allows anyone to get creative with technology. BBC has donated a free micro:bit to every 11- or 12-year-old child in year 7 or equivalent across the United Kingdom.

The BBC micro:bit is based on a Nordic nRF51822 MCU with 16K RAM and 256K Flash. There’s also an onboard accelerometer and magnetometer from Freescale.

Features

- Can be programmed with high-level online IDEs using the BBC’s website at http://www.microbit.co.uk/create-code including:
  - Microsoft TouchDevelop IDE
  - Microsoft Blocks
  - CodeKingdoms Javascript
  - MicroPython
- mbed enabled
  - Online IDE at developer.mbed.org
  - Easy to use C/C++ SDK
  - Dedicated micro:bit runtime libraries for rapid development (developed by Lancaster University)

![Pinout](https://os.mbed.com/platforms/Microbit/)

**Figure 1.13** The BBC Micro:bit development board and its pinout from the Arm® Mbed™ website. (Source: https://os.mbed.com/platforms/Microbit/)
Introduction to Arm® Mbed™

- Nordic nRF51822 multi-protocol Bluetooth® 4.0 low energy/2.4GHz RF SoC
  - 32-bit ARM® Cortex M0 processor (16MHz)
  - 16 kB RAM
  - 256 kB Flash
  - Bluetooth Low Energy Master/Slave capable
- Input/Output
  - 25 LED matrix
  - Freescale MMA8652 3-axis accelerometer
  - Freescale MAG3110 3-axis magnetometer (e-compass)
  - Push Button ×2
  - USB and Edge connector serial I/O
  - 2/3 reconfigurable PWM outputs
  - 5 x banana/croc-clip connectors
  - Edge connector
  - 6 x analog in
  - 6-17 GPIO (configuration dependent)
  - SPI
  - i2c
- USB Micro B connector
- JST power connector (3v)

Further Information about Micro:bit
https://www.microbit.co.uk/
https://os.mbed.com/platforms/Microbit/

1.4.4 The Arm® Mbed™ Ethernet Internet of Things (IoT) Starter Kit

This Ethernet IoT Starter Kit includes an Arm® Mbed™ Freedom FRDM-K64F development board and mbed application shield (Figure 1.14). It is designed for the IBM IoT Foundation and is aimed to provide the user with a slick experience. It allows the user to send data from the onboard sensors into the IBM cloud easily. It is particularly suitable for developers with no specific experience in embedded or web development, as it provides a platform for learning new concepts and creating working prototypes. It allows the user to access to IBM cloud applications through IBM’s BlueMix platform, in which deployment and device management are very simple. The starter kit hardware can also be modified and extended to suit specific needs.

The FRDM-K64F development board is the next-generation development board. It uses a power-efficient Kinetis K64F MCU featuring an ARM® Cortex®-M4 core running up to 120 MHz and embedding 1024 KB Flash, 256 KB RAM, and lots of peripherals (16-bit ADCs, DAC, timers) and interfaces (Ethernet, USB device crystal-less, and serial). The new mbed application shield has been designed to enable the maximum number of potential experiments with Arduino form factor development boards, keeping as much in common with the mbed application board as possible.

This book focuses on the Arm® Mbed™ IBM Ethernet IoT Starter Kit. The Arm® Mbed™ Ethernet IoT Kit contents:
Mbed Enabled NXP K64F Development Board

- NXP K64F Kinetis K64 MCU (MK64FN1M0VLL12)
- High-performance ARM\textsuperscript{R} Cortex\textsuperscript{TM}-M4 Core with floating point unit and DSP
- 120 MHz, 256 KB RAM, 1 MB flash

**mbed Application Shield**

- 128×32 graphics LCD
- 5-way joystick
- 2 × potentiometers
- Speaker, PWM connected
- 3-axis +/-1.5 g accelerometer
- RGB LED, PWM connected
- Temperature sensor
- Socket for XBee (ZigBee) or RN-XV (WiFi)

**MCU Features**

- Kinetis MK64FN1M0VLL12 in 100LQFP
- Performance
  - ARM\textsuperscript{R} Cortex\textsuperscript{TM}-M4 32-bit core with DSP instructions and floating point unit (FPU)
  - 120 MHz max CPU frequency
- Memories and memory interfaces
  - 1024 KB program flash memory

*Figure 1.14 The Arm\textsuperscript{R} Mbed\textsuperscript{TM} Ethernet Internet of Things (IoT) Starter Kit, which includes a FRDM-K64F development board (left) and its application shield (right).*
Introduction to Arm® Mbed™

- 256 KB RAM
- FlexBus external bus interface

**System peripherals**
- Multiple low-power modes, low-leakage wake-up unit
- 16-channel DMA controller

**Clocks**
- 3× internal reference clocks: 32 KHz, 4 MHz, and 48 MHz
- 2× crystal inputs: 3–32 MHz (XTAL0) and 32 kHz (XTAL32/RTC)
- PLL and FL

**Analog modules**
- 2× 16-bit SAR ADCs up 800 ksp (12-bit mode)
- 2× 12-bit DACs
- 3× analog comparators
- Voltage reference 1.13 V

**Communication interfaces**
- 1× 10/100 Mbit/s Ethernet MAC controller with MII/RMII interface IEEE1588 capable
- 1× USB 2.0 full-/low-speed device/host/OTG controller with embedded 3.3V/120mA Vreg, and USB device crystal-less operation
- 1× Controller area network (CAN) module
- 3× SPI modules
- 3× I2C modules. Support for up to 1 Mbit/s
- 6× UART modules
- 1× Secure digital host controller (SDHC)
- 1× I2S module

**Timers**
- 2× 8-channel Flex-Timers (PWM/Motor control)
- 2× 2-channel Flex-Timers (PWM/Quad decoder)
- 32-bit PITs and 16-bit low-power timers
- Real-time clock (RTC)
- Programmable delay block

**Security and integrity modules**
- Hardware CRC and random-number generator modules
- Hardware encryption supporting DES, 3DES, AES, MD5, SHA-1, and SHA-256 algorithms

**Operating characteristics**
- Voltage range: 1.71 to 3.6 V
- Flash write voltage range: 1.71 to 3.6 V

**Board Features**

**Onboard components**
- FXOS8700CQ—6-axis combo sensor accelerometer and magnetometer
- 2 user push buttons
- RGB LED

**Connectivity**
- USB full-/low-speed on-the-go/host/device controller with on-chip transceiver, 5 V to 3.3 V regulator and micro-USB connector
– Ethernet 10/100 controller with onboard transceiver and RJ45 connector
– Up to 5× UARTs, 2× SPIs, 2× I2Cs and 1× CAN connected to headers (multiplexed peripherals)

● Extensions
– Micro SD-card socket
– Headers compatible with Arduino R3 shields (32-pins / outer row)
– Headers for proprietary shields (32-pins / inner row)

● Analog and digital IOs (multiplexed peripherals)
– Up to two ADC 16-bit resolution with 28 analog I/O pins connected to headers
– Up to three timers with 18 PWM signals accessible from headers
– Up to six comparator inputs or one DAC output
– Up to 40 MCU I/O pins connected to headers (3.3 V, 4 mA each, 400 mA max total)

● Board power-supply options (onboard 5 to 3.3 V regulator)
– USB debug 5 V
– USB target 5 V
– 5–9 V Vin on Arduino headers
– 5 V PWR input
– Coin-cell 3.3 V

● Integrated open SDA USB debug and programming adapter
– Several industry-standard debug interfaces (PEmicro, CMSIS-DAP, JLink)
– Drag-n-drop MSD flash programming
– Virtual USB to serial port

● Form factor: 3.2” × 2.1” / 81 mm × 53 mm

● Software development tools
– mbed HDK & SDK enabled
– Online development tools
– Easy-to-use C/C++ SDK
– Lots of published libraries and projects
– Alternate offline options NXP free KDS (compiler toolchain) and KSDK library/examples

● Supplier website: http://www.nxp.com/frdm-k64F

Figure 1.15 shows the FRDM-K64F development board’s component layout and pinout. Following are the most used pins:

RGB LED LED1 (LED_RED), LED2(LED_GREEN), LED3 (LED_BLUE), LED4 (LED_RED)

Digital inputs/outputs D0, D1, D2, ..., D15
Analog inputs A0, A1, A2, A3, A4, A5
Analog outputs DAC0_OUT
PWM (pulse width modulation) A4, A5, D3, D5, D6, ..., D13

Further Information about FRDM-K64F

https://os.mbed.com/platforms/IBMEthernetKit/
https://os.mbed.com/platforms/FRDM-K64F/
https://os.mbed.com/components/mbed-Application-Shield/
http://www.nxp.com/frdm-k64F
1.5 Summary

This chapter first explains what an embedded system is and discusses the difference between microcontrollers and microprocessors. It then introduces the ARM® architecture and Arm® Mbed™ systems.

1.6 Chapter Review Questions

Q1.1 What is an embedded system?

Q1.2 What is the difference between microcontrollers and microprocessors?

Q1.3 How does CPU work?

Q1.4 Use a suitable diagram to describe ARM® Processor Architecture.

Q1.5 What is Arm® Mbed™? Describe the concepts of mbed cloud services, clients, and mbed OS.

Q1.6 Use a table to compare the key features between LPC1768 and FRDM-K64F.

Figure 1.15 FRDM-K64F development board’s component layout (left) and the Arduino and NXP header pinout from the Arm® Mbed™ website. (Source: https://os.mbed.com/platforms/FRDM-K64F/)