CSE190 Fall 2023
Lecture 3
MCUs (cont.)

Wireless Embedded Systems
Aaron Schulman
How to choose an MCU for a project?

- What metrics we need to consider?
  - Power consumption
  - Clock frequency
  - I/O
  - Memory
  - Internal functions
  - Size
How to choose MCU for our project?

- What metrics we need to consider?
  - Power consumption
    - E.g., we cannot afford high-power MCU because the power budget of the system requires lasting two years on one battery charge.
  - Clock frequency (speed that instructions are executed)
    - kHz is too slow...
    - 100MHz is over kill...
  - I/O
    - Lots of peripherals you can have:
      Image sensor, UART debugger, SD card, DAC, ADC, microphone, LED
How to choose MCU for our project?

• What metrics we need to consider?
  – Memory
    • We need to have sufficient memory to store:
      – Program (Non-volatile): Logic to read from sensors, communicate
      – Stack: Function calls are now expensive (no recursion)
      – Data: Constants (time periods), Sensor history, Communication state
        » We may need non-volatile data storage for data too (e.g., Flash)
  – Performance of internal peripherals
    • E.g., Speed of copying data from the sensor to the radio (DMA)
How to choose MCU for our project?

- **Memory**
  - Store accelerometer history data
    - 12bits each for X,Y,Z acceleration (36 bits)
    - sampled 2 thousand times a second (2 KHz)
    - = 36*2,000 bits per second (72 kbit/s or 9 kByte/s)
    - How many seconds can we hold if we have only 100 kBytes of storage?
  - What types of memory are available on an MCU?
    - Internal memory: SRAM, 0.5~128 kBytes, non-volatile FRAM also available
    - External memory: Flash, high power consumption, ~5mA for read and ~10mA for erase
How to choose MCU for our project?

• Clock frequency
  – kHz is too slow
    • Smartphone camera frame rate is 60fps
      (1 KHz clock would leave only 60 clock cycles per frame)
  – 100MHz is too fast
    • Power consumption is high
      (power increases linearly with clock speed)
  – O(10) MHz is ideal for most embedded applications
How to choose MCU for our project?

• I/O (interface for external peripherals)
  – Interfacing sensors, debugger, LEDs, Bluetooth radio
  – Every I/O needs physical pins on the chip
  – We often need a large number of I/O pins
  – We need various types of I/O pins
    (some pins can do more than one function)
  • Analog pins (input/output analog signals e.g., audio)
  • Digital pins (input/output digital signals e.g., busses, GPIOs)
The MCU used in our projects ($10)

<table>
<thead>
<tr>
<th>Core Processor</th>
<th>ARM® Cortex®-M4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Core Size</td>
<td>32-Bit Single-Core</td>
</tr>
<tr>
<td>Speed</td>
<td>80MHz</td>
</tr>
<tr>
<td>Connectivity</td>
<td>CANbus, EBI/EMI, I²C, IrDA, LINbus, MMC/SD, QSPI, SAI, SPI, SWPMI, UART/USART, USB OTG</td>
</tr>
<tr>
<td>Peripherals</td>
<td>Brown-out Detect/Reset, DMA, PWM, WDT</td>
</tr>
<tr>
<td>Number of I/O</td>
<td>82</td>
</tr>
<tr>
<td>Program Memory Size</td>
<td><strong>1MB (1M x 8)</strong></td>
</tr>
<tr>
<td>Program Memory Type</td>
<td>FLASH</td>
</tr>
<tr>
<td>EEPROM Size</td>
<td>-</td>
</tr>
<tr>
<td>RAM Size</td>
<td>128K x 8</td>
</tr>
<tr>
<td>Voltage - Supply (Vcc/Vdd)</td>
<td>1.71V ~ 3.6V</td>
</tr>
<tr>
<td>Data Converters</td>
<td>A/D 16x12b; D/A 2x12b</td>
</tr>
<tr>
<td>Oscillator Type</td>
<td>Internal</td>
</tr>
<tr>
<td>Operating Temperature</td>
<td>-40°C ~ 85°C (TA)</td>
</tr>
<tr>
<td>Mounting Type</td>
<td>Surface Mount</td>
</tr>
<tr>
<td>Package / Case</td>
<td>100-LQFP</td>
</tr>
<tr>
<td>Supplier Device Package</td>
<td>100-LQFP (14x14)</td>
</tr>
<tr>
<td>Base Product Number</td>
<td>STM32L475</td>
</tr>
</tbody>
</table>