Lecture 3: Overview of the MSPM0 Microcontroller

ME/AE 6705

Introduction to Mechatronics

Dr. Jonathan Rogers

Georgia Tech

Lesson Objectives

- Understand the basic components of the TI MSPM0 MCU evaluation board
  - And know where major components are located
- Be able to explain the major components of the MCU and evaluation board and their purpose
- Be able to explain basic performance characteristics of MSPM0

TI MSPM0 Evaluation Kit

- Texas Instruments makes a broad range of industrial-grade microcontrollers

Low-Power MCU's

Real-Time Control

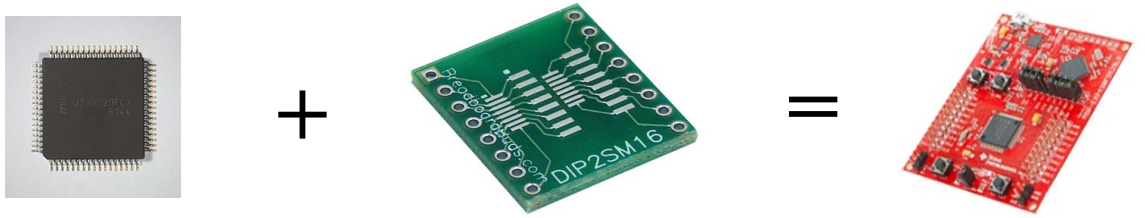
Wireless/Internet Connectivity

Georgia Tech

TI MSPM0 Evaluation Kit

- MCU is just “processor” that looks like integrated circuit (IC) with dozens of pins
  - To use, either need to build your own “breakout board” to attach wires to pins
  - Or buy an “evaluation/development board” where someone has already done this for you

MCU + Breakout Board = Dev/Evaluation Kit



Texas Instruments’ line of development kits are called ***Launchpads***.

TI MSPM0 Evaluation Kit

- Microcontroller pinout for MSPM0

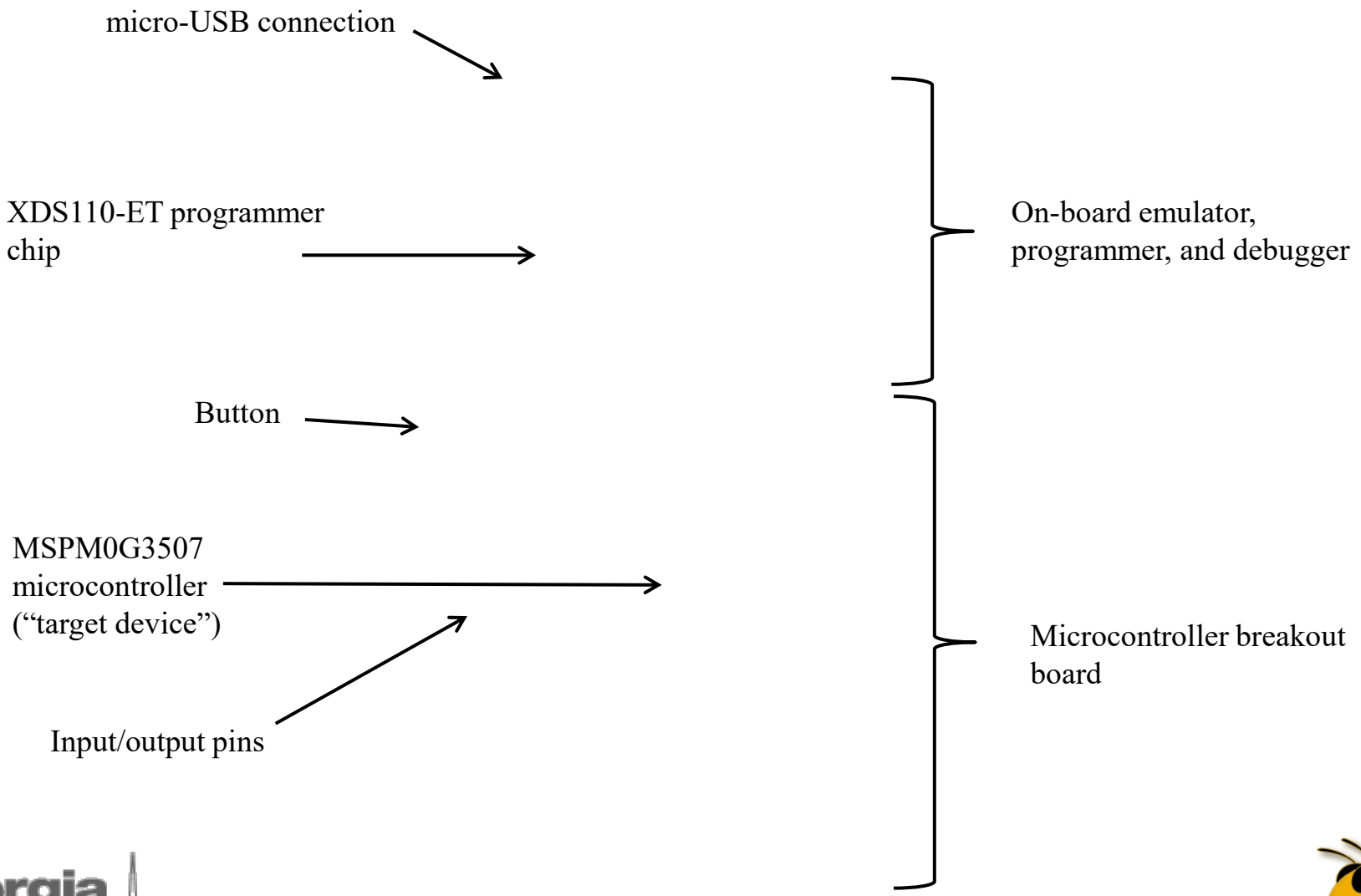
Georgia Tech

TI MSPM0 Launchpad

- For this course, we have chosen to use the MSPM0 series microcontroller
  - From Low-power + Performance line of MCU's
  - 32-bit processor for good computing performance
  - Reasonable input-output capabilities
  - Costs around $15-$20
  - Similar to what is found in hand-held or battery powered devices

Georgia Tech

TI MSPM0 Launchpad



Georgia Tech

TI MSPM0 Launchpad

- Device split into two basic components:
  - On-board “emulator” – allows programming and debugging
  - Target device – actual microcontroller and breakout board

TI MSPM0 Launchpad

- MSPM0G3507 microcontroller
  - Low power ARM M0+ 32-bit processor
  - Up to 80 MHz system clock
  - 128 KB flash, 32 KB SRAM
  - Two 16-bit timers, one 32-bit timers
  - Four serial communication (UART) ports
  - Two 12-bit analog-to-digital converters (ADC)
  - One 12-bit digital-to-analog converter (DAC)

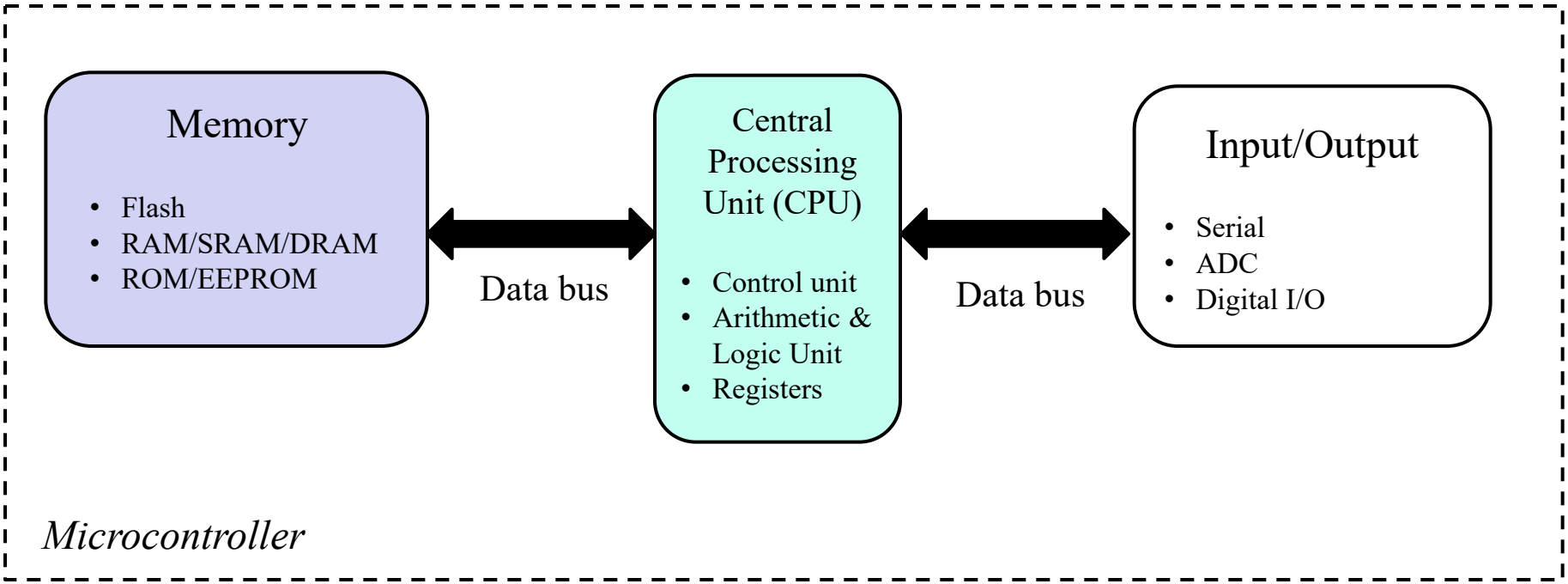
ARM: type of processor architecture that requires significantly fewer transistors compared to "complex instruction set computing" x86 processors

⟶ Reduces cost, heat, and power requirements!

Georgia Tech

MSPM0G3507 Microcontroller

- **Microprocessors** are complex IC's that interface with memory and other I/O components on bus (wires)
- **Microcontrollers** are IC's that incorporate memory and I/O components on single chip



MSPM0G3507 Microcontroller

- All microcontrollers have a central processing unit (CPU) that executes program instructions.
- CPU's contain three main elements:
  - **Control Unit**: Generates timing signal used to fetch program instruction from memory and execute it
  - **Arithmetic and Logic Unit**: Performs actual instruction on data (such as addition of two numbers)
  - **Registers**: Memory locations inside CPU that hold internal data while instructions are being executed

MSPM0G3507 Microcontroller

- Memory structures:
  - **Volatile memory**: Data is maintained in memory as long as power is applied. If power removed, data will be lost.
  - **Non-volatile memory**: Data is maintained even when power is removed.

- In PC's, memory obviously different from storage space
  - Volatile memory in PC's is typical RAM
  - Where is non-volatile memory on PC located?

MSPM0G3507 Microcontroller

- Microcontrollers have very specialized memory types (more so than typical PC's)

Memory Type	Description	
ROM	Read Only Memory	Nonvolatile memory that is programmed during manufacture of chip. Data can be read but cannot be written during use. Used for fixed programs (pre-installed libraries, pre-installed programs, etc).
PROM	Programmable ROM	Same as ROM but can be programmed once by user with no further changes allowed.
EPROM	Erasable PROM	Can be programmed more than once. Contents can be erased by shining UV light through window on top of device.
EEPROM or Flash memory	Electrically-Erasable PROM	Similar to EPROM, but contents can be erased by applying high-voltage signal rather than UV light.
RAM	Random Access Memory	Volatile memory that requires power to operate. Access time for data is constant and is not dependent on physical location of data.
DRAM	Dynamic RAM	RAM that uses capacitors to store data. Data must be refreshed periodically (rewritten) due to charge leakage.
SRAM	Static RAM	RAM in which data does not need to be refreshed as long as power is applied. Faster than DRAM but more expensive.

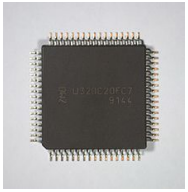
MSPM0G3507 Microcontroller

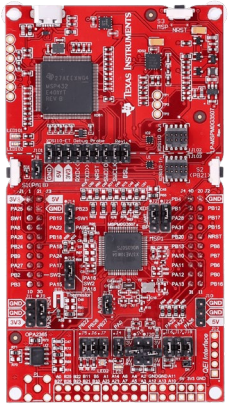
- Memory on this device:
  - 128 KB flash
    - *Some used to store program, some available for user to store data*
  - 32 KB SRAM
    - *Used to store variables and data structures in program*

MSPM0G3507 Microcontroller

MSPM0G3507

- Up to 80 MHz clock (32-bit)
- 32 KB SRAM
- 128 KB flash (non-volatile storage)
- Emphasis on compactness, low cost, low power





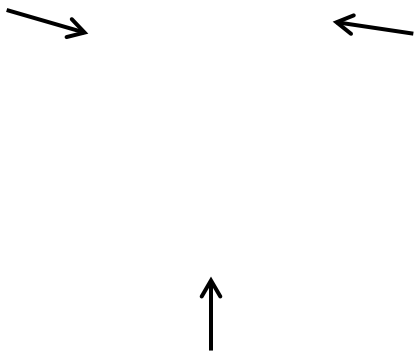
MacBook Pro (base model)

- 2.6 GHz clock (64-bit)
- 8 GB SRAM
- 500 GB solid state hard disk drive
- Emphasis on high speed, large word size

Georgia Tech

General Purpose Input / Output

- Key feature of microcontrollers is their incorporation of input and output pins directly on chip



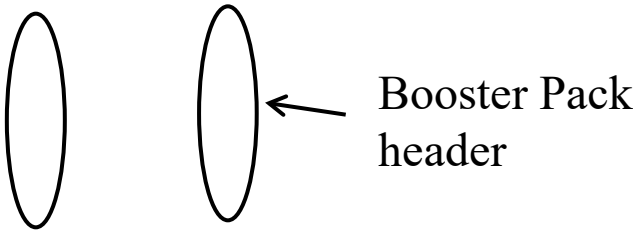
These input and output pins serve a variety of functions (some pins serve multiple functions) including:

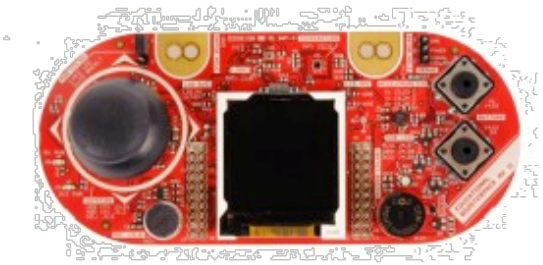
- Provide power to the microcontroller
- Analog to digital conversion – take a reading of a voltage and convert to an equivalent number for use by CPU
- Digital input or output – read whether a pin is high or low, or write a pin high or low (to turn on LED for instance)
- Serial communication (UART, I^2C, SPI) – communicate more complex data with external device
- Pulse Width Modulation – write specialized signal for actuator control

Georgia Tech

General Purpose Input / Output

- Many microcontroller boards can be paired with additional peripheral boards
  - Usually called “shields”
  - For TI Launchpads, called “Booster Packs”
- These come with additional sensors and are made to plug into pin headers on evaluation board





TI Educational BoosterPack MLII for MSP Launchpad

General Purpose Input / Output

- What do all these pins do? Pinouts can be found in User Guide.
  - Note some pins have multiple functions – you can select their functionality during programming process

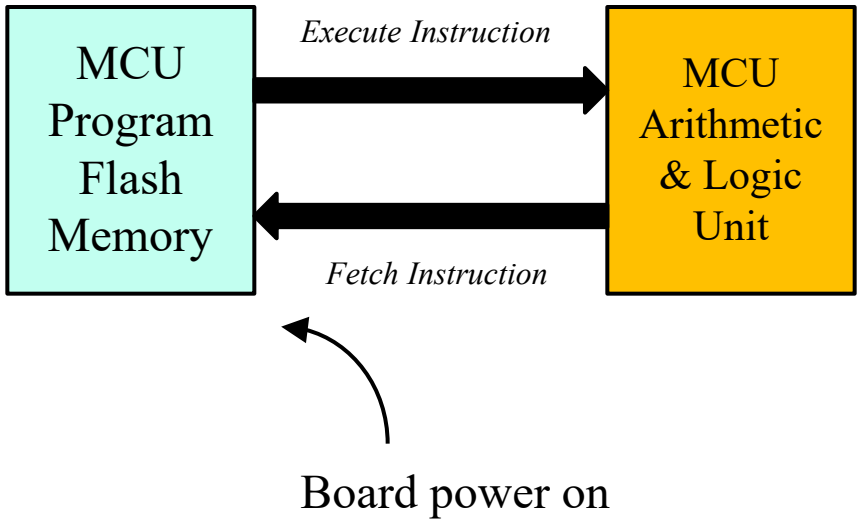
General Purpose Input / Output

- Board includes 3 buttons and several LED's
  - These are simply additional input (buttons) and output (LEDs) pins just like any other

We will have lecture on GPIO to cover these topics in more detail.

Programming MCUs

- Workflow for microcontroller operation



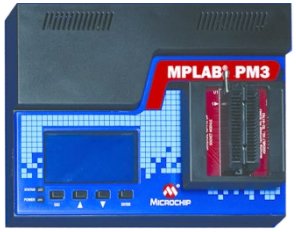
- As soon as board starts up, it reaches into program memory (flash) and loads its first instruction you programmed it with
- Thus your program executes every time *on start up*
- Your program continues to run until board powered off
- Unlike typical PC, *board does nothing except what you program it to do!*

Georgia Tech

Programming MCUs

- Also unlike PC's, you cannot write your code on MCU
- Instead, you write code on PC and then "program" it into flash memory on MCU
  - MCU must be connected to external programmer
  - When MCU restarts, it will run code that was flashed to it
- External programmers are needed for most MCUs

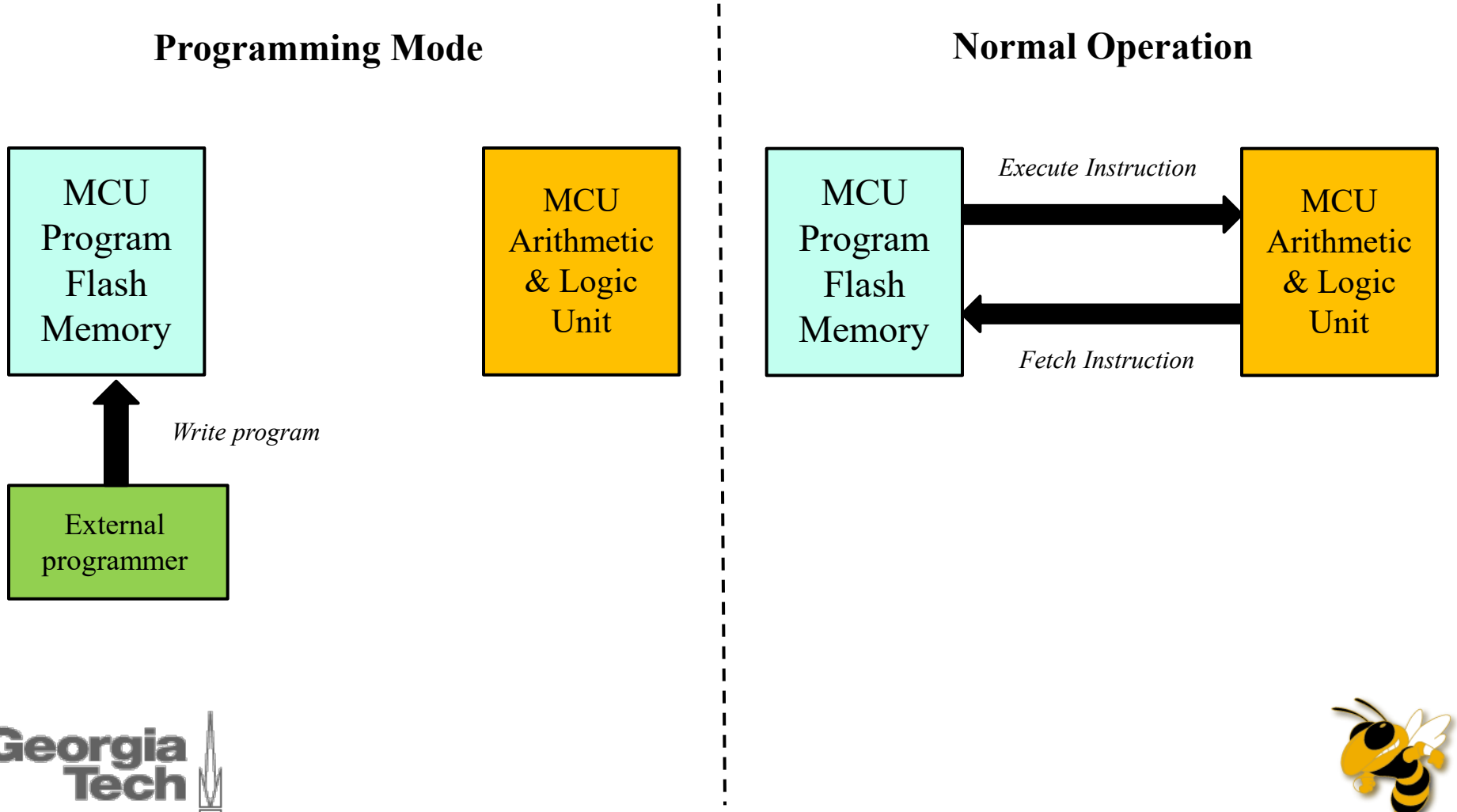
Microchip Programmer for PIC MCUs



STMicro Programmer for STM8/STM32 MCUs

Georgia Tech

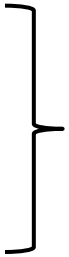
Programming MCUs



Georgia Tech

Programming the MSPM0

- MSPM0 Launchpad comes with on-board programmer
  - This is basically what is on top half of board
  - Called XDS110-ET on-board emulator



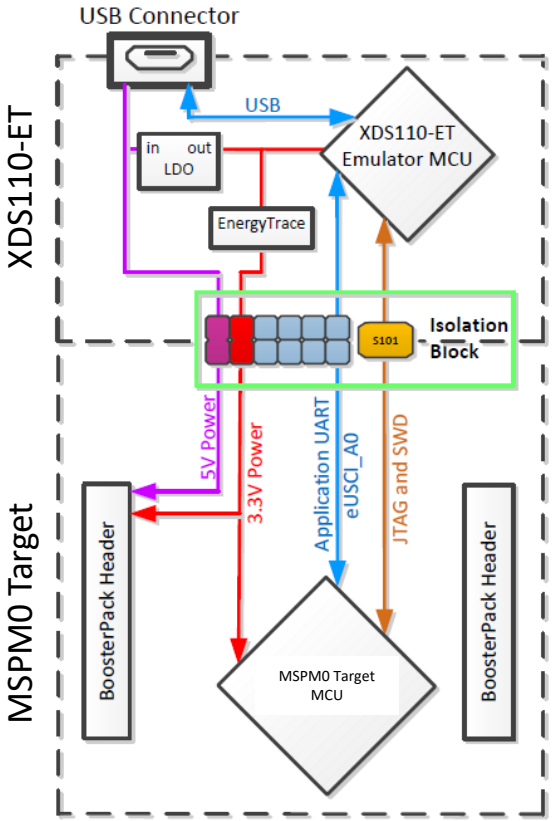
XDS110 On-board emulator

Purpose of on-board emulator:

- Provide USB connection to PC
- Program MCU
- Enable PC-based debugging of MCU code
- Provide power to MCU from USB power connection

On-board Emulator Isolation Block

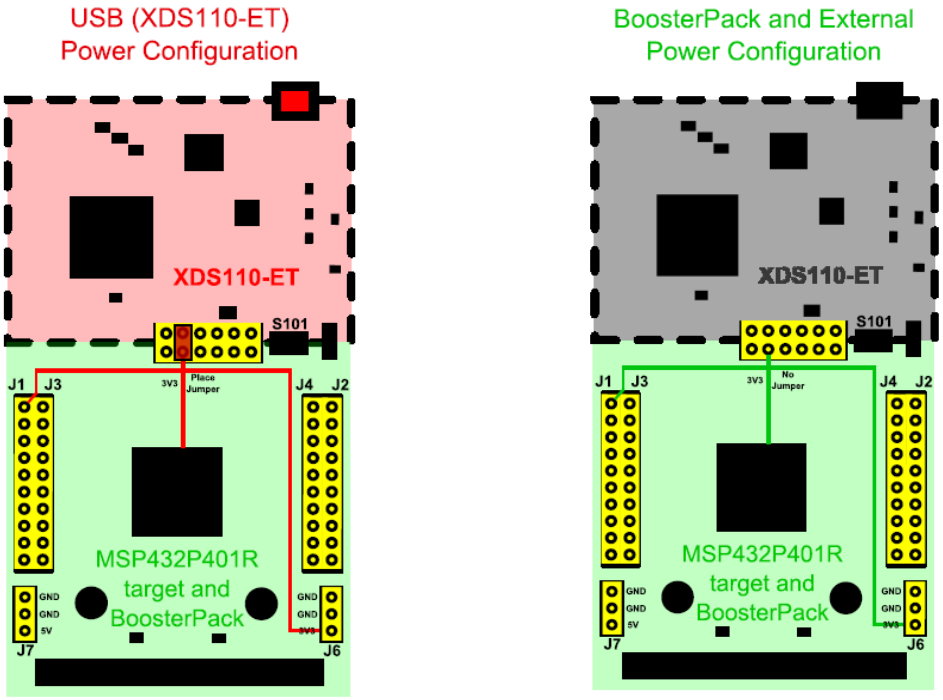
- One purpose of XDS is to power MCU directly from USB (coming from computer)
- MCU runs off of 3.3V DC. Jumper connections provide 3.3V and 5V to MCU block across “Isolation Block”
- Isolation block is group of jumpers that connect MCU part of board to emulator part of board



Georgia Tech

On-board Emulator Isolation Block

- When 3.3V jumper is in place, power provided to board via USB
- When 3.3V jumper removed, external power (3.3 VDC) must be applied to board via pin J1.1 (+V) and J3.22 (GND)



On-board Emulator Isolation Block

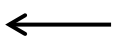
- Note regarding 5V signal:
  - USB from computer provides 5V to board
  - When 3.3V jumper is connected, this 5V is stepped down to 3.3V in emulator
  - Isolation block allows 3.3V and 5V to be passed through, which are then available on several output pins on board

On-board Emulator Isolation Block

- Another important function of emulator is to provide communication to computer via USB
  - Used for programming, debugging, streaming data to and from computer, etc.
  - Universal Asynchronous Receiver/Transmitter (UART) serial communication used for this purpose
  - We will learn about this in future lecture
  - Rest of jumpers in isolation block are used for UART communication with PC
    - *TXD and RXD are transmit and receive lines*
    - *RTS and CTS are hardware flow control lines (not typically used)*

Device Reset

- Oftentimes when debugging you may want to reboot the device to restart your program
- This is accomplished using the NRST button (S1)
- NRST is directly connected to the NRST pin on the MSPM0G3507 MCU (pin 38) on datasheet



Georgia Tech

Documentation

- Four **very important** sources of documentation on MSPM0G3507 Launchpad

1. MSPM0G3507 Datasheet
   - Describes microcontroller only
   - Provides lots of details on MCU hardware and packaging
   - Is primary source we will use for this class to understand features of our MCU
   - Available on Canvas under Resources tab

Documentation

2. LP-MSPM0G3507 Launchpad Users Guide
   - Describes evaluation kit
   - Provides less detail, more high level overview
   - Describes features of breakout board and emulator, not related to MCU directly
   - Is another primary source we will use for this class to understand features of our MCU
   - Also available on Canvas under Resources tab

Documentation

3. MSPM0 G-Series Technical Reference Manual
   - Provides all details on all MCU subsystems
   - Describes register functions and programming guidelines
   - Contains all information associated with MCU
   - 1,607 pages
   - Available on Canvas under Resources tab

Documentation

4. MSPM0G1X0X-G3X0X Driver Library Users Guide
   - Documents Driver Library APIs for programming MCU
   - Contains functions definitions, variable names, etc.
   - Found at this link:

https://dev.ti.com/tirex/explore/node?node=A__AKc6sRDImfLcqfXkLIu3AQ__MSPM0-SDK__a3PaaoK__LATEST

Question 1

- Using the pin map in the Launchpad kit, determine:
  1. Which pins you would use to for receive and transmit if you are implementing UART communication with an external device.
  2. Where these pins are located on the board?

Question 2

- Suppose you want to power the MCU externally (from a battery). How would you go about doing so?
  - Where do you connect power?
  - What voltage should you use?
  - What other steps are involved?

Example Program

- Flash LED's starting at ~1 HZ
- Button press (S1) toggles oscillation between 1 Hz and ~5 Hz
- Demonstrate use of RST button

Georgia Tech