Advanced LabVIEW

http://workshop.frclabviewtutorials.com

Customizing the Dashboard

Customizing the Dashboard

- Open Project

Customizing the Dashboard

- Open Project
- Sending data from robot

Customizing the Dashboard

- Open Project
- Sending data from robot
  - Smart Dashboard VI’s
  - Named (case sensitive) values

Customizing the Dashboard

- Open Project
- Sending data from robot
  - Smart Dashboard VI’s
  - Named (case sensitive) values

Customizing the Dashboard

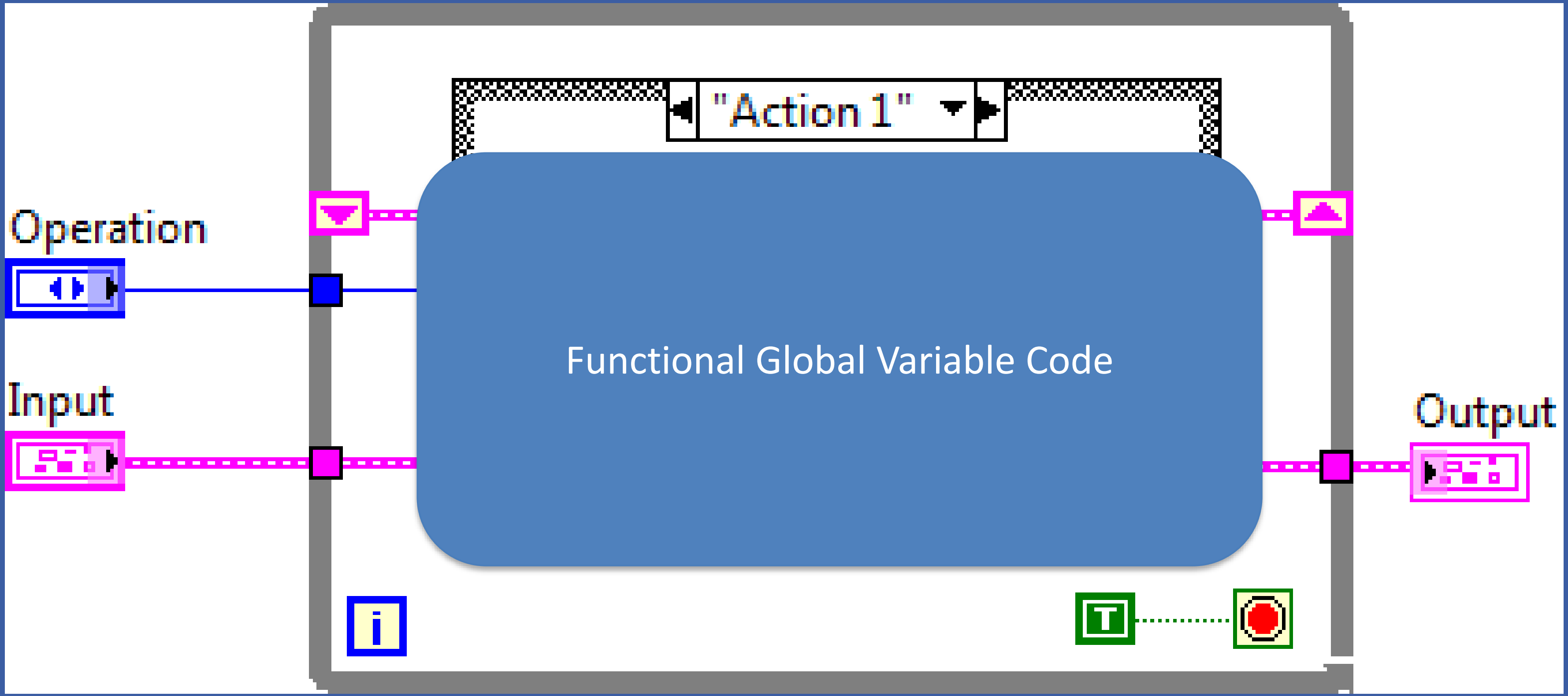
- Open Project
- Sending data from robot
- Sending data to robot

Functional Global Variable

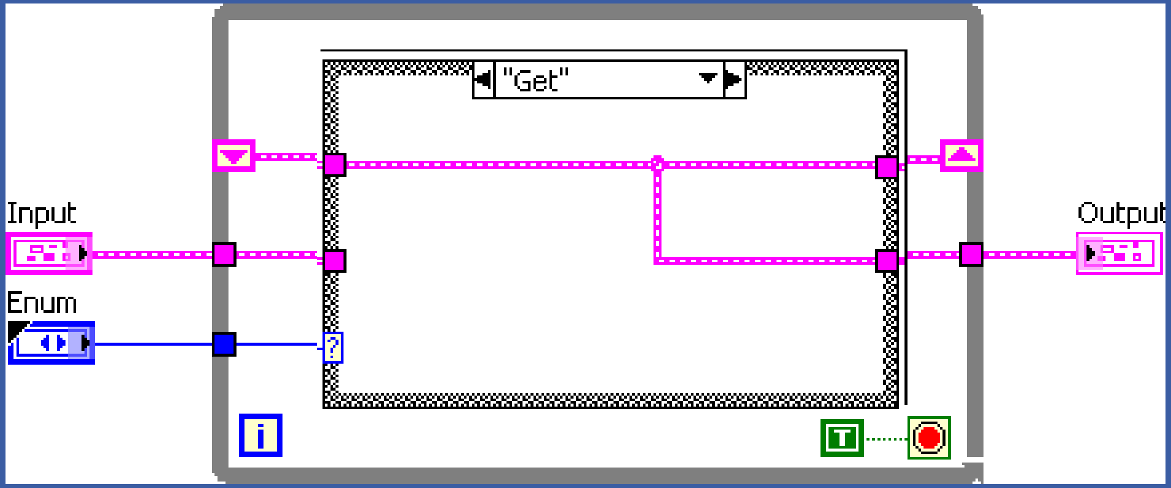
Functional Global Variable

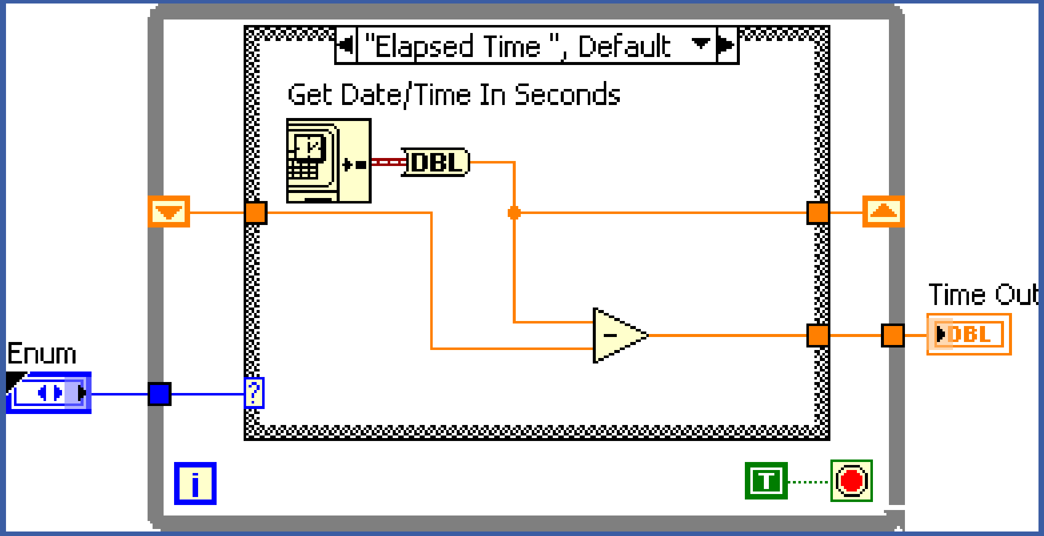
- Side note
  - https://frclabviewtutorials.com/tutorials/fgv/

FGV


"Action 1"
Operation
Input
Functional Global Variable Code
Output

Implementing An FGV





VI Properties

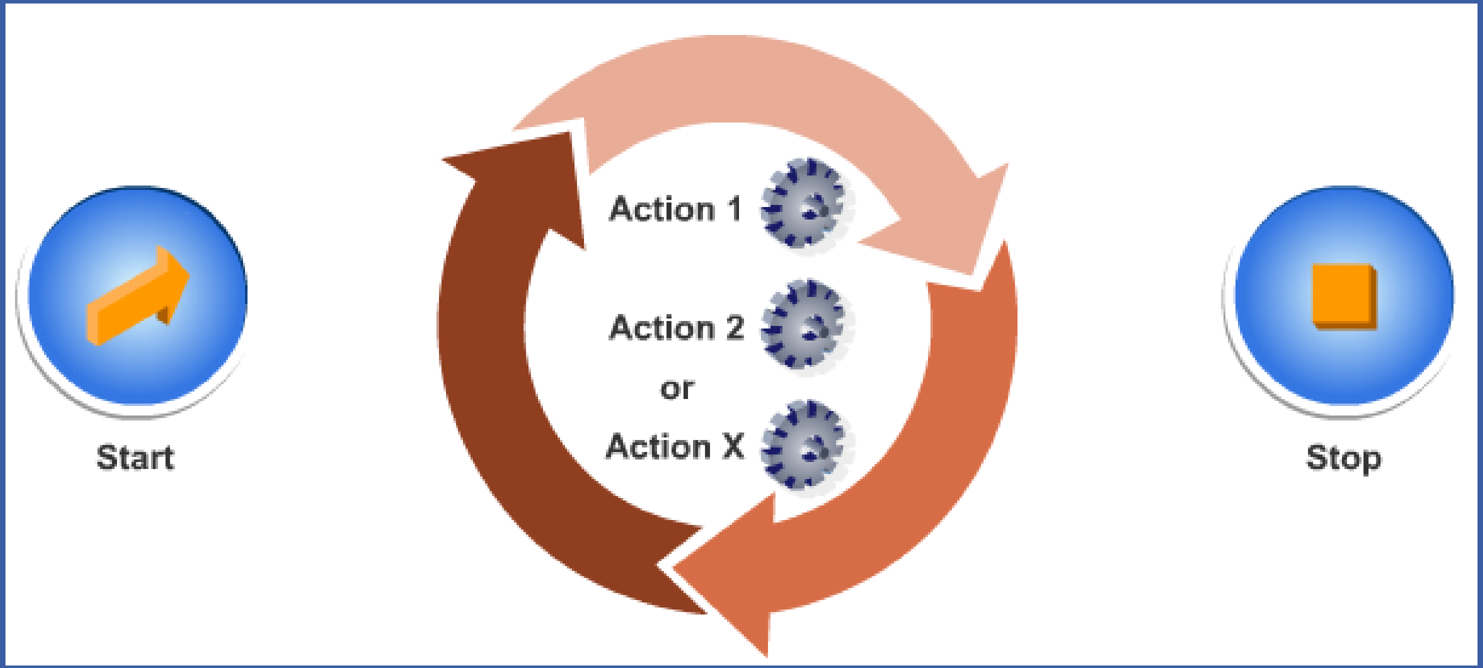
- Quick Intro
  - https://frclabviewtutorials.com/tutorials/fgv/
- SR Flip Flop Demo

VI Properties

- Quick Intro
  - https://frclabviewtutorials.com/tutorials/fgv/
- SR Flip Flop Demo
  - Edge Detector
- https://frclabviewtutorials.com/tutorials/memory-library/

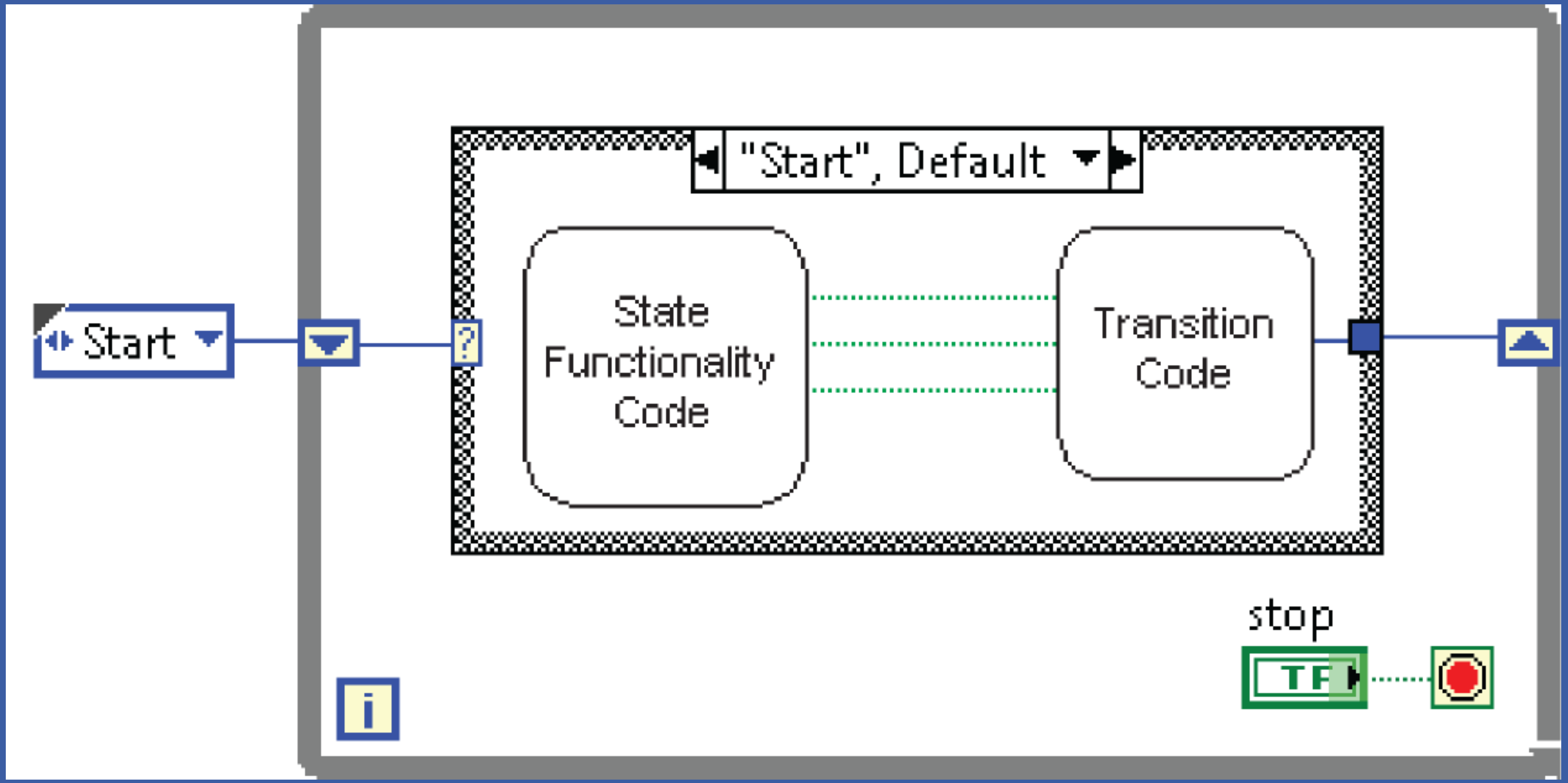
Architectures

- State Machine



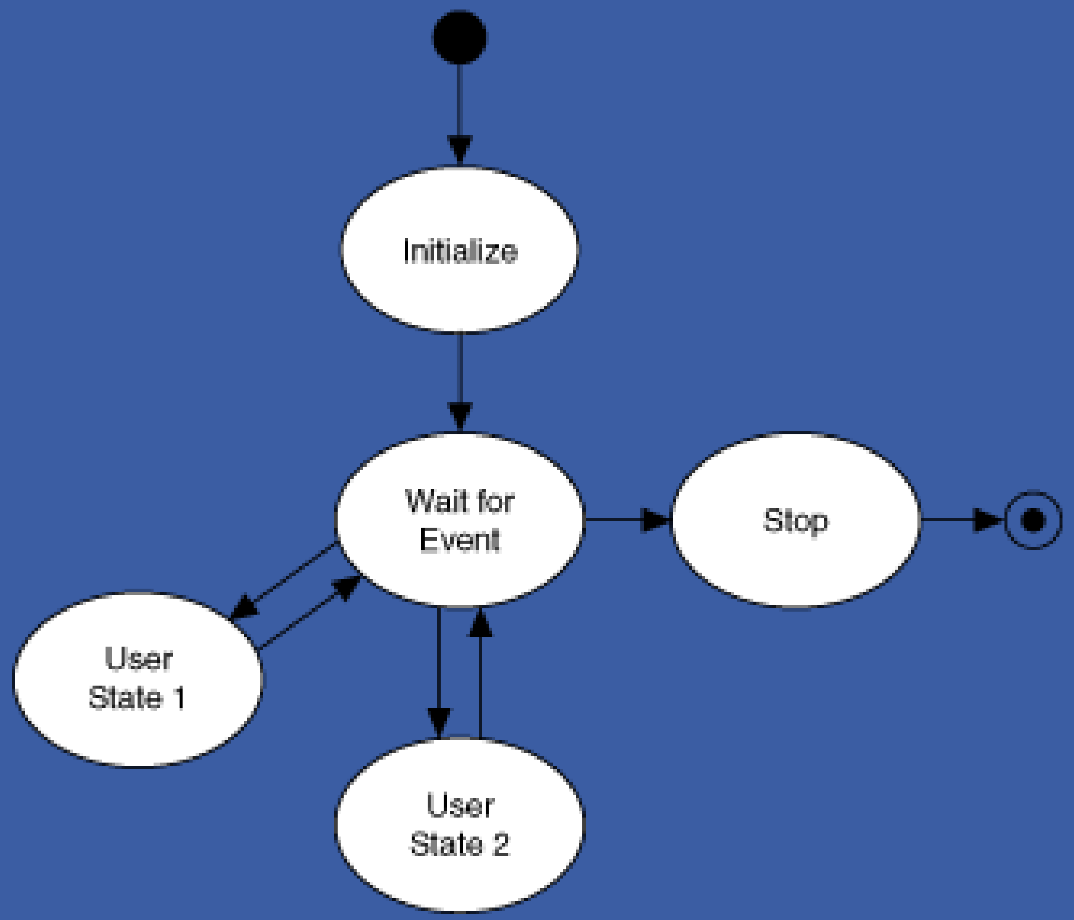
Architectures

- State Machine



Architectures

- State Machine



Architectures

- State Machine
- Producer-Consumer
  - Parallel loops
    - First creating data or instructions
    - Other handling

Architectures

- State Machine
- Producer-Consumer
  - Parallel loops
  - Use either queue or fgv

Producer Consumer Demo

Producer Consumer Demo

- (side note)
  - In Computer Science (and CE, but software specifically), there's a concept call "separation of concerns" (Wikipedia: link)
  - [Each segment of code should only deal with a single task] (paraphrased)
    - This might be:
      - Getting input
      - Or controlling the shooter

This set-up, allows you to separate the task of deciding what to do base on inputs (/auto) and how to do it(/interacting with the hardware)

Type Def.

- Useful for passing data – both controls and indicators
- Demo

Type Def.

- Useful for passing data – both controls and indicators
- Demo

Closed loop control through PID

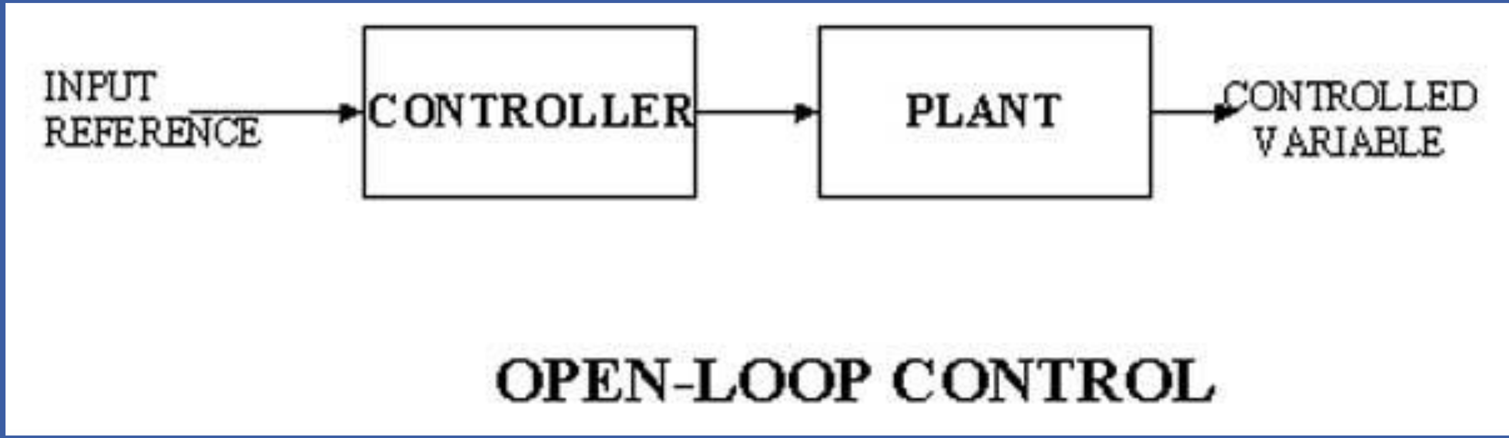
PID

PID

- WPI Video:
  - https://www.youtube.com/watch?list=PL8BLGj0RyhMzNXX9gHBosWPRbqqn0gJUQ&v=UOuRx9Ujsog&feature=emb_logo

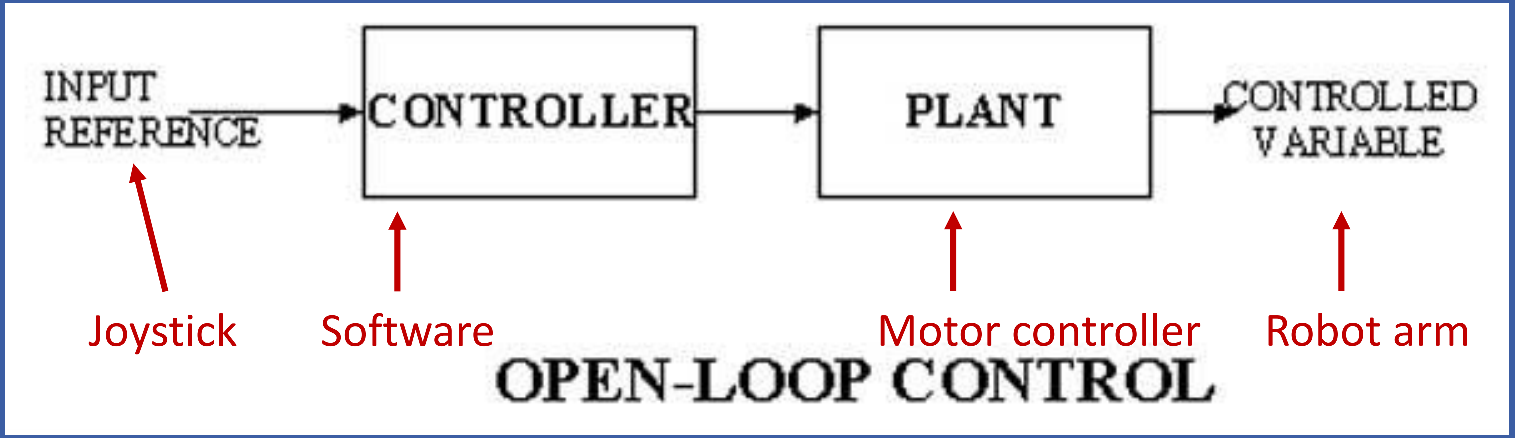
Closed Loop Control

- Open Loop:



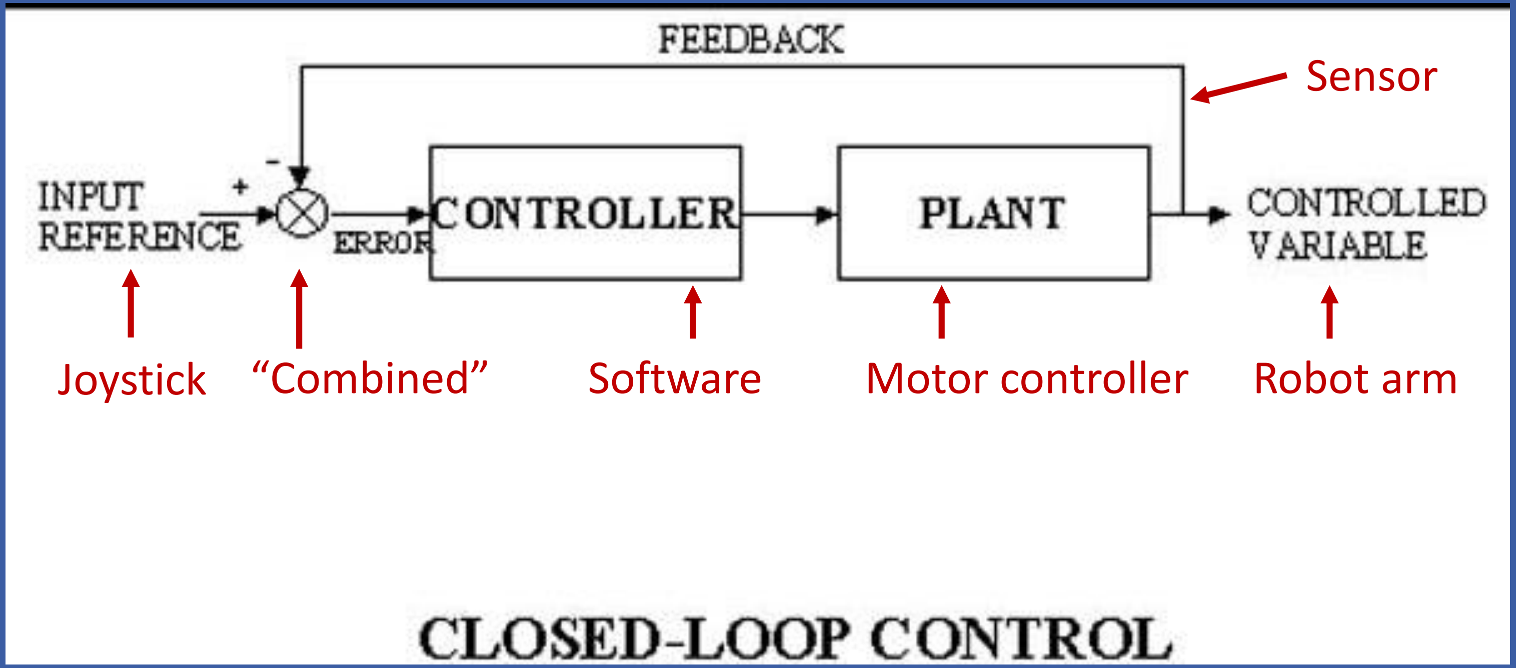
Closed Loop Control

- Open Loop:



Closed Loop Control

- Open Loop
- Closed Loop



Closed Loop Control

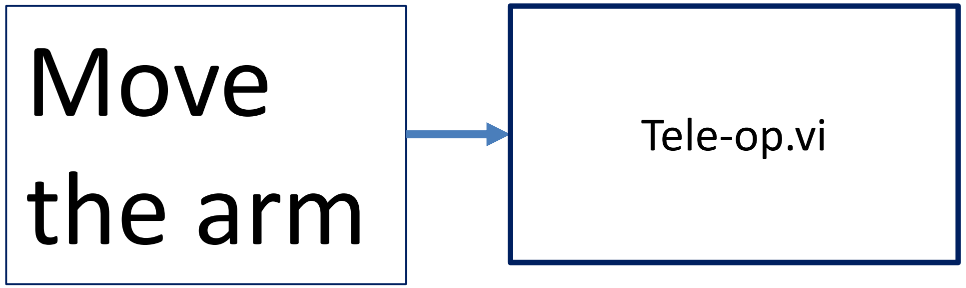
- Open Loop
- Closed Loop
  - Example

Closed Loop Control

- Open Loop
- Closed Loop
  - Example

Closed Loop Control

- Open Loop
- Closed Loop
  - Example

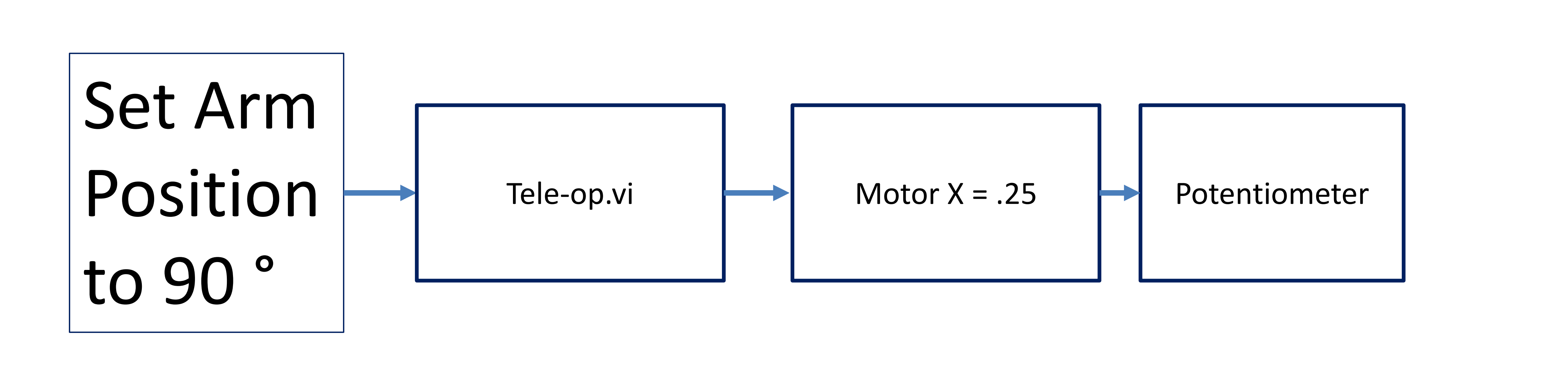


Closed Loop Control

- Open Loop
- Closed Loop
  - Example

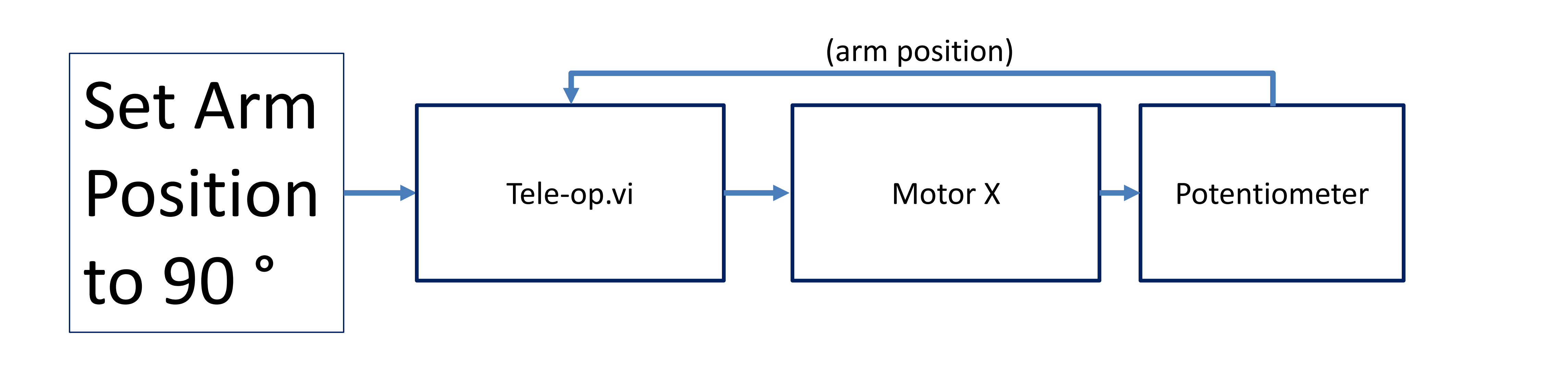
Closed Loop Control

- Open Loop
- Closed Loop
  - Example



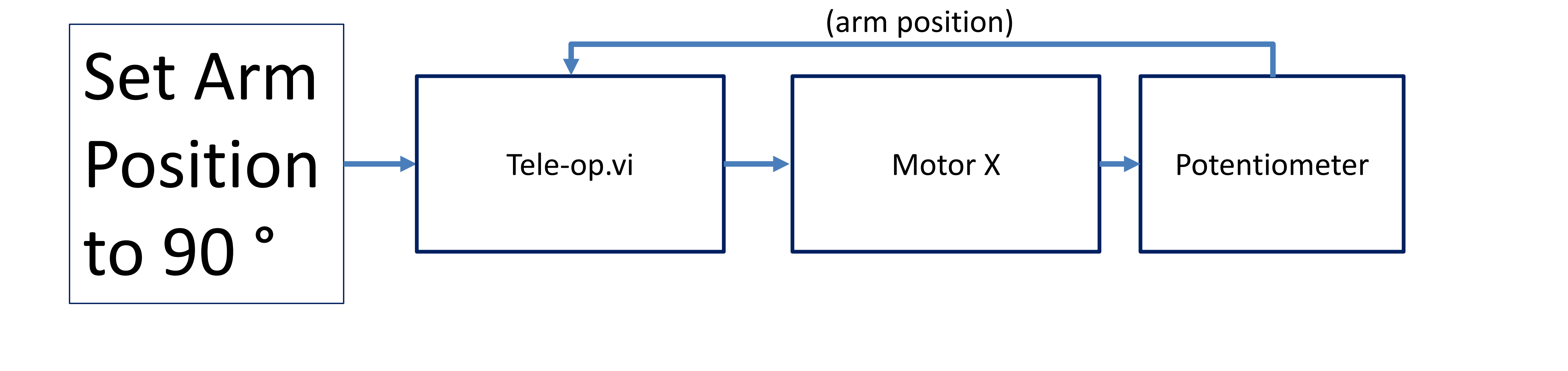
Closed Loop Control

- Open Loop
- Closed Loop
  - Example



Closed Loop Control - PID

- PID stand for:
  - Proportional
  - Integral
  - Derivative



Closed Loop Control - PID

- PID stand for:
  - Proportional
  - Integral
  - Derivative

$$Output = K_p\, E(t) + K_i \int E'(t) \; + K_d \; E'(t)$$

PID

- Proportional

PID

- Proportional
  - Constant multiplied by error (offset)
  - The larger this is, the faster the robot approaches the setpoint (smaller rise time)
  - If too large, the robot will overshoot the target consistently

PID

- Proportional
  - Constant multiplied by error (offset)
  - The larger this is, the faster the robot approaches the setpoint (smaller rise time)
- Integral
  - Constant multiplied by integral of all previous error values
  - Used to eliminate steady state error (reducing offset after movement)
  - If too large, robot will eventually (> 5s) respond vehemently

PID

- Proportional
  - Constant multiplied by error (offset)
  - The larger this is, the faster the robot approaches the setpoint (smaller rise time)
- Integral
  - Constant multiplied by integral of all previous error values
  - Used to eliminate steady state error (reducing offset after movement)
- Differential
  - The larger this is, the less overshoot and settling time (less bounce)
  - If too large,

PID

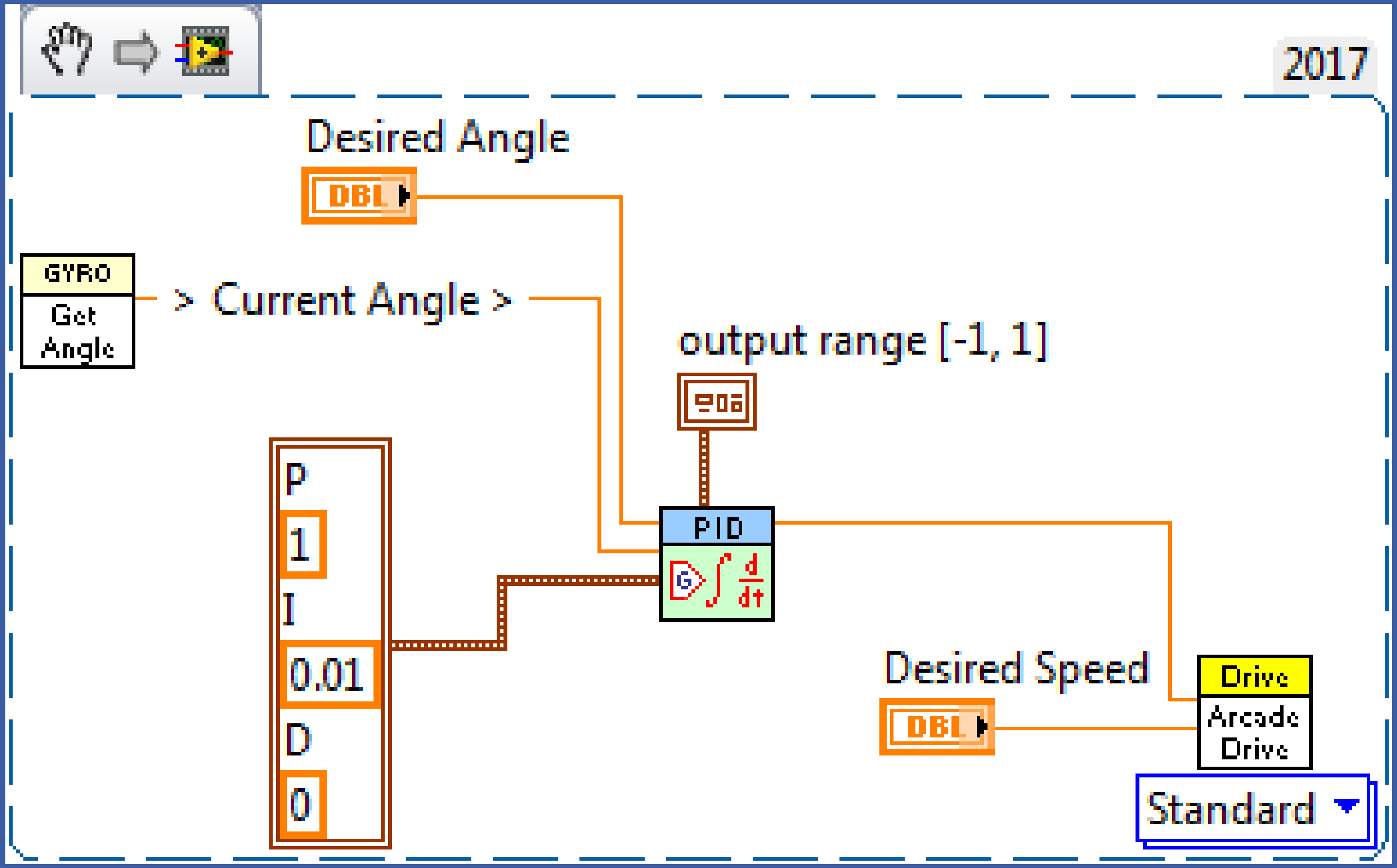
- Tuning

PID

- Tuning
  - Several methods available
    - **Ziegler–Nichols***
    - **Tyreus Luyben**
    - **Cohen–Coon**
    - **Åström-Hägglund**
    - **Manual Tuning***

PID

- Example code



Questions