

# PERIPHERAL DEMO

By: Anthony Kennedy, Patrick Mock, Mitchell Turton, Chinmay  
Bandapalli, Landon D Slater

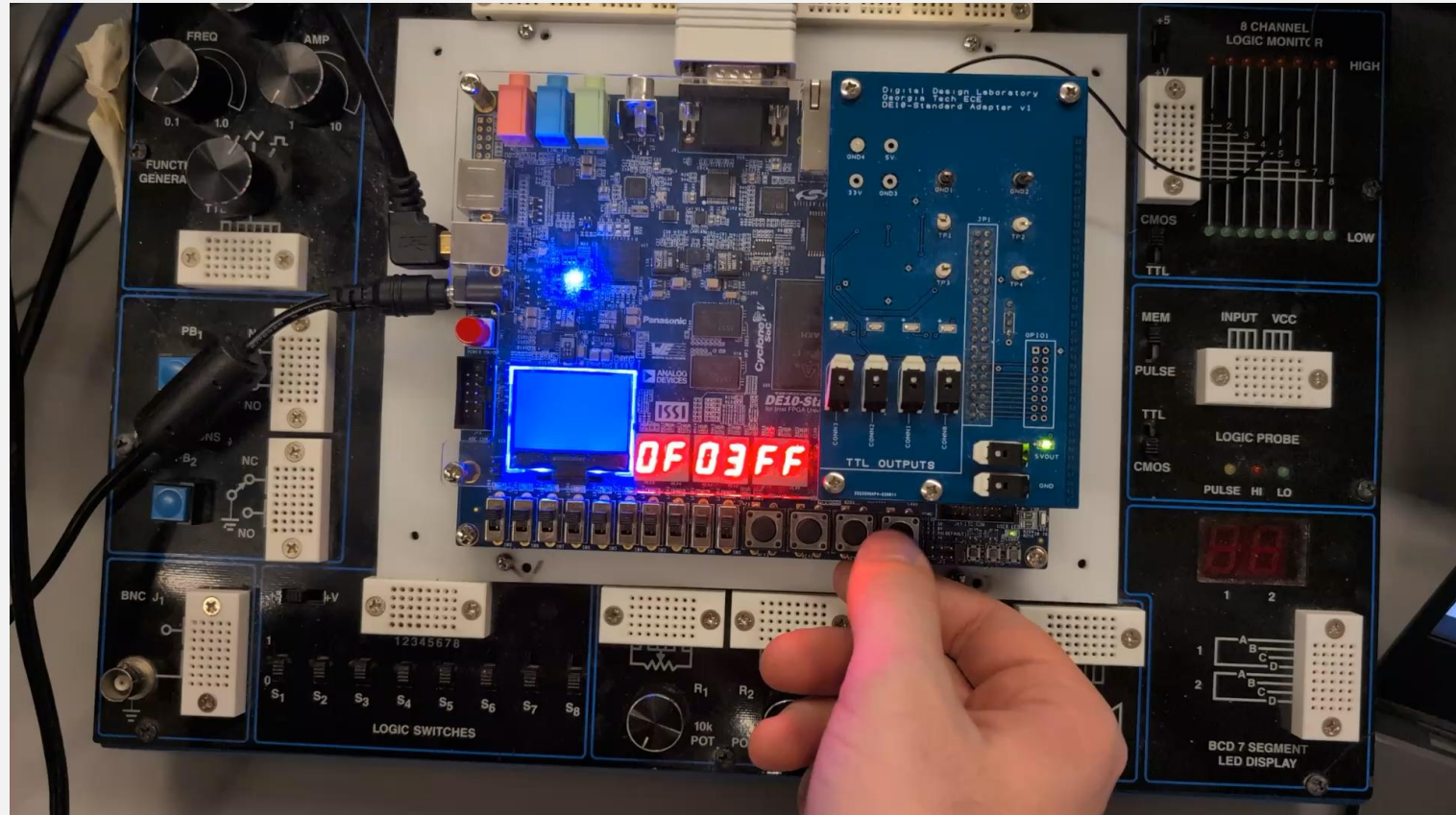
# INTRODUCTION

- Goal: Create a peripheral for the DE-10 Board to control the 10 LEDs with an easy to use interface for an SCOMP programmer
  - Our Ideology: "Flexibility and Ease of Use Over All Else"

**FUNCTIONALITY**

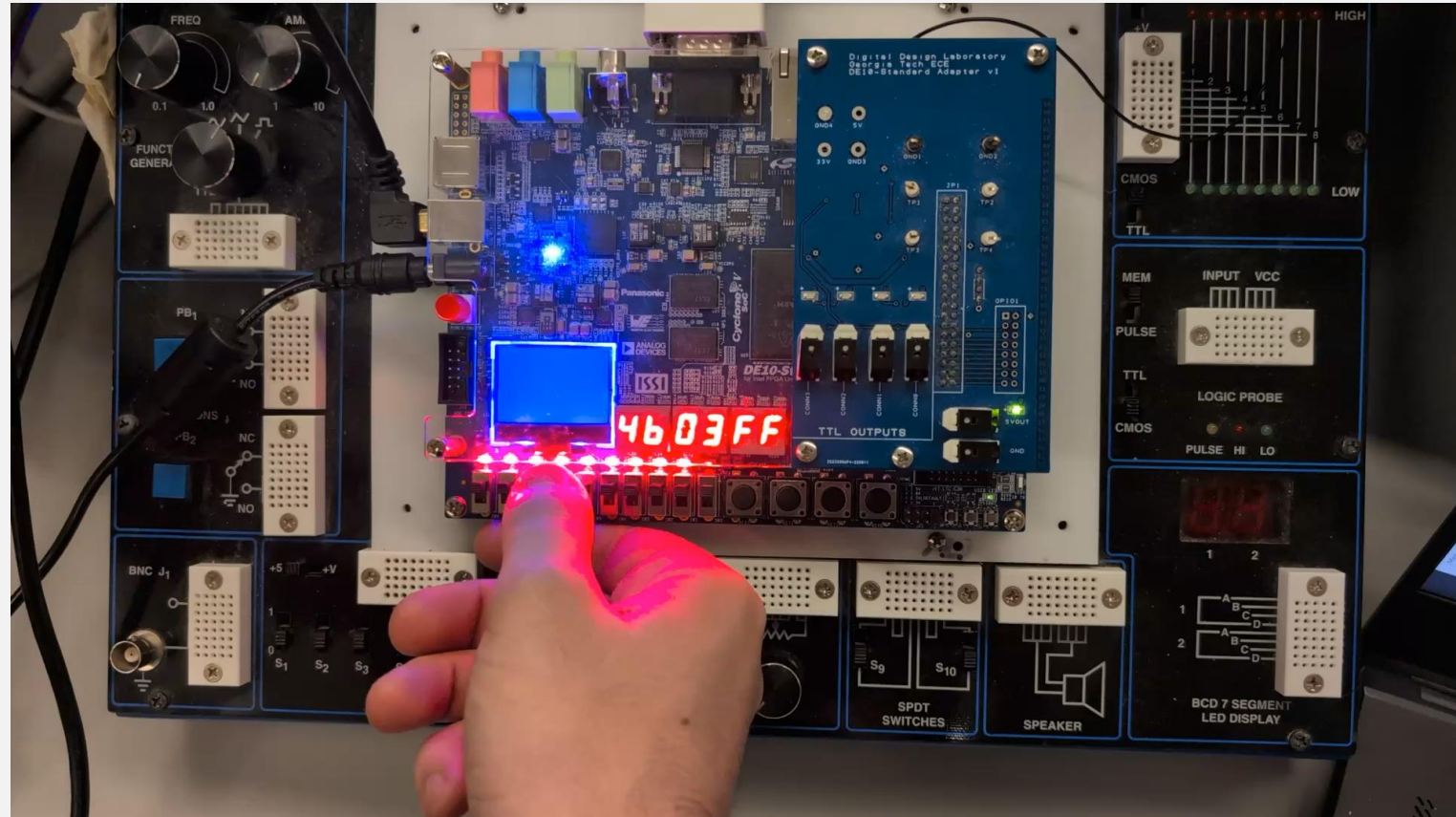
# FUNCTIONALITY: INDIVIDUAL BRIGHTNESS CONTROL

- 256 Brightness Levels
- Gamma Correction
- Set All Brightness Functionality



# FUNCTIONALITY: STATE REGISTER BIT MASK

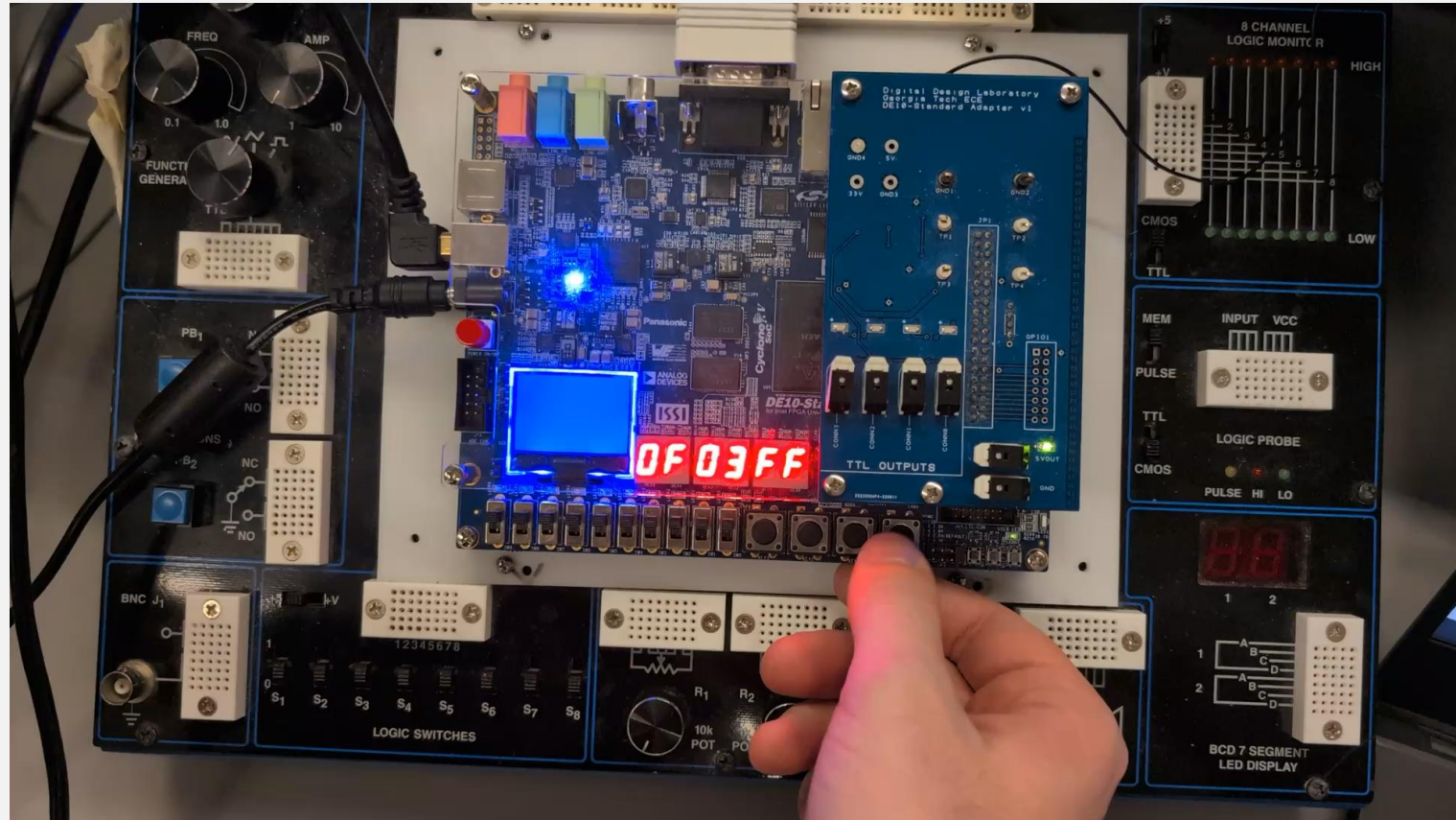
- Sets if each LED is 'on' or 'off'
- Programmer easily able to display digital binary information





# FUNCTIONALITY: REGISTER READ FUNCTIONALITY

- Quality of life feature
- Allows for cleaner more scalable code



## API

Register	What it Stores	Default/Reset Value
ST_REG [10-bit] (r/w)	10-bit string serving as bit mask	Reset → 0
LED_SEL [4-bit] (r/w)	Value 0-9 selects what BR_CTL controls	Reset → 0
BR_CTL [16-bit] (r/w)	Lower 8 bits used to control brightness level of selected LED	Reset → 255 [All Br_Regs]
BR_CTL_ALL [8-bit] (w)	Sets all LEDs to this brightness value when written to	Reset → N/A

# EASE OF USE



```

LOADI 5
STORE LED_Sel_MEM ; Memory address reserved for LED_Sel
OUT LED_Sel
CALL Inc_LED_Pair_10_NO_READ

; Increment LED Brightness Pair by 10 without Read
Inc_LED_Pair_10_NO_READ:
    LOAD LED_Sel_MEM
    ADDI LED_Array_Base_Tag
    STORE LED_PTR
    LOAD LED_PTR

    ADDI 10
    OUT BR_Ctl
    STORE LED_PTR

    LOAD LED_Sel_MEM
    ADDI 1
    OUT LED_Sel
    STORE LED_Sel_MEM

    ADDI LED_Array_Base_Tag
    STORE LED_PTR
    LOAD LED_PTR

    ADDI 10
    OUT BR_Ctl
    STORE LED_PTR

    RETURN

LED_PTR: DW 0

LED_Sel_MEM: DW 0

LED_Array_Base_Tag:
    ; [10 lines allocating variable for Brightness]

```

Without Read:



```

LOADI 5
OUT LED_Sel
CALL Inc_LED_Pair_10_READ

; Increment LED brightness by 10
Inc_LED_Pair_10_READ:
    IN BR_Ctl
    ADDI 10
    OUT BR_Ctl

    IN LED_Sel
    ADDI 1
    OUT LED_Sel

    IN BR_Ctl
    ADDI 10
    OUT BR_Ctl

    RETURN

```

With Read:

Ease of Use: Read Functionality

```

; Set all brightnesses to zero
LOADI 0
STORE i
Loop:
    LOAD i
    ADDI -10
    JZERO End
    LOAD i
    OUT LED_SEL
    LOADI 0
    OUT BR_CTRL
    LOAD i
    ADDI 1
    STORE i
    JUMP Loop
End:

```

Without BR\_CTL\_ALL:



```

; Set all brightnesses to zero
LOADI 0
OUT BR_CTL_ALL

```

With BR\_CTL\_ALL:

Ease of Use: BR\_CTL\_ALL

IN CONCLUSION