



Sensors & Microsystem Electronics: microcontrollers

PART 3: SIMULATOR

Atmel studio simulator

- Use the simulator to:
 - Discover the specific operation of an instruction, block of instructions, macro, or function
 - Verify the flow of execution
 - Find out what causes unintended or erratic behaviour

Atmel Studio simulator

SIMULATOR

- A **program** (software) **mimics** the execution of the instructions and the state of the microcontroller
- Execution of instructions stops at predefined breakpoints
- The simulator updates the emulated registers, settings, memory cells, ...

Simulation time \neq run time on the microcontroller

! Simulating 1 second of the program can take several minutes !

I/O can be viewed and influenced with the checkboxes in the simulator

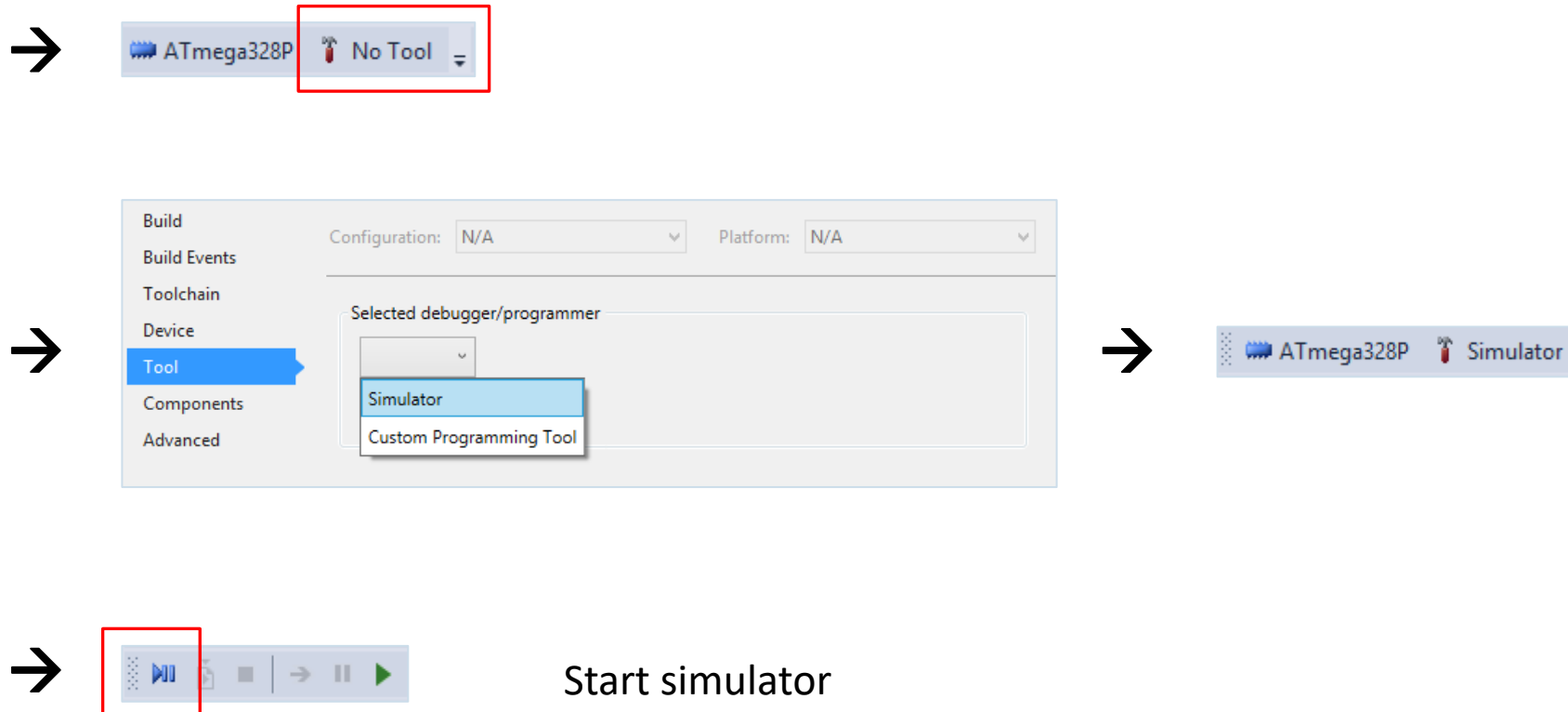
DEBUGGER

- The **microcontroller** itself **executes** the instructions
- Execution of instructions stops at predefined breakpoints
- The state of registers, settings, memory cells, etc. is communicated to Atmel Studio at breakpoints via serial communication

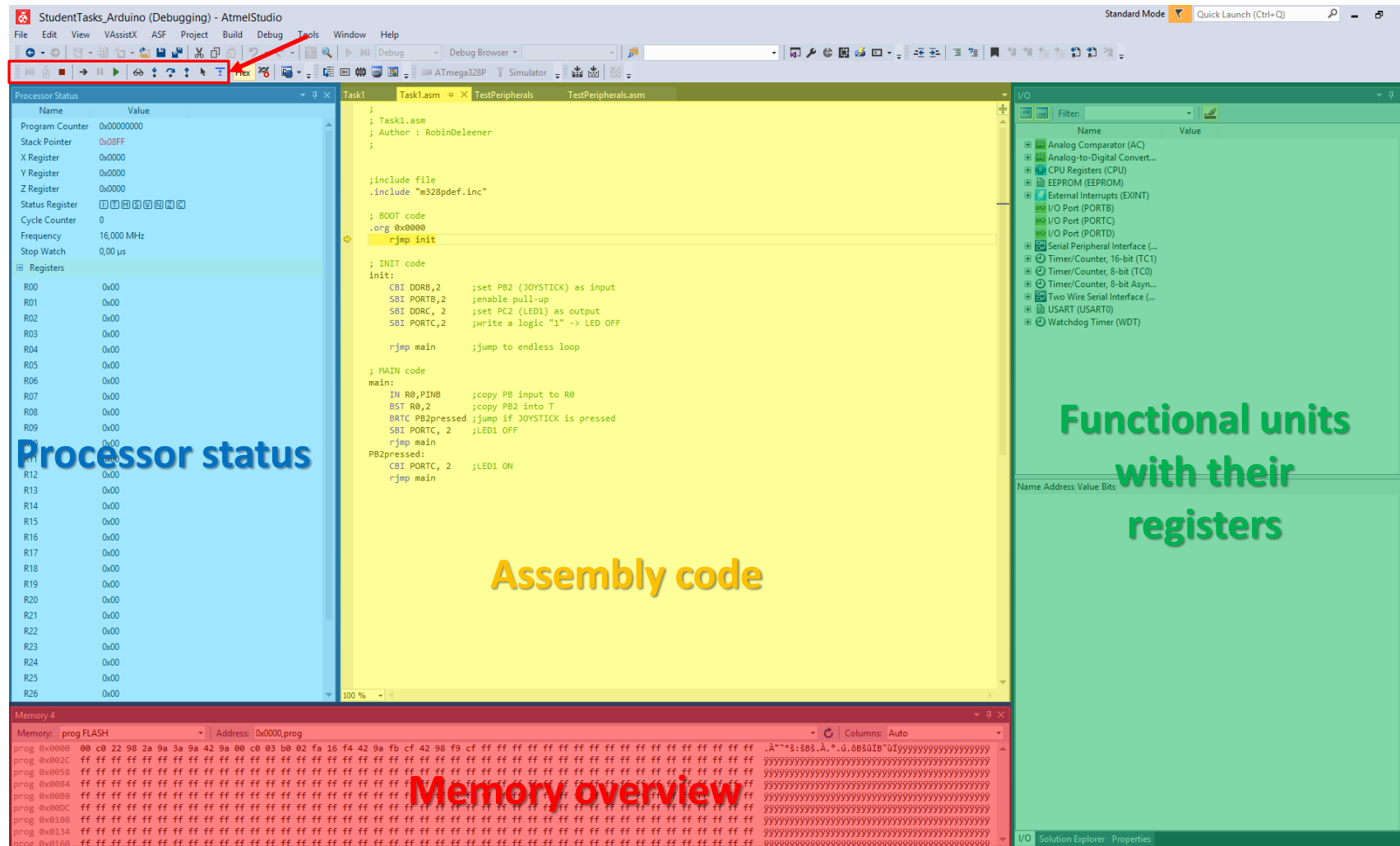
Peripherals influence the program flow and the state of the microcontroller directly

Atmel studio simulator

Select the simulator



Atmel studio simulator

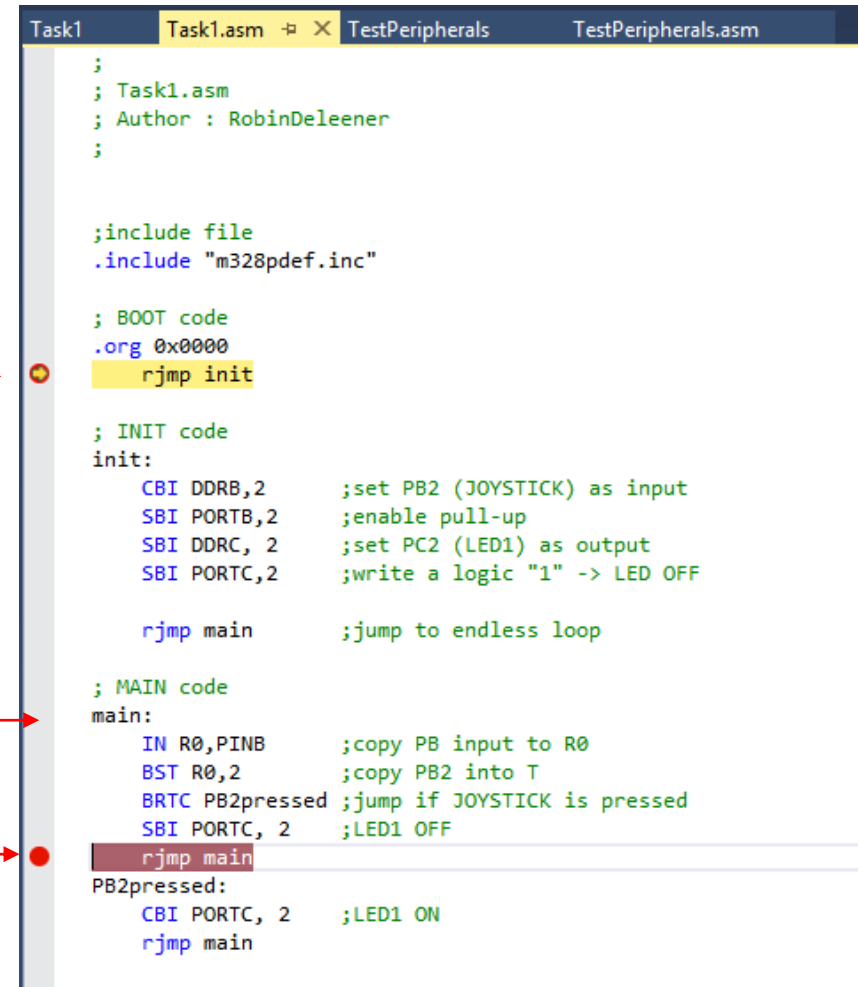


Atmel studio simulator

Assembly code

Yellow arrow marks the instruction that will be executed

Double click in the left blue bar to set or clear a breakpoint



```
Task1 Task1.asm [X] TestPeripherals TestPeripherals.asm
;
; Task1.asm
; Author : RobinDeleener
;

;include file
.include "m328pdef.inc"

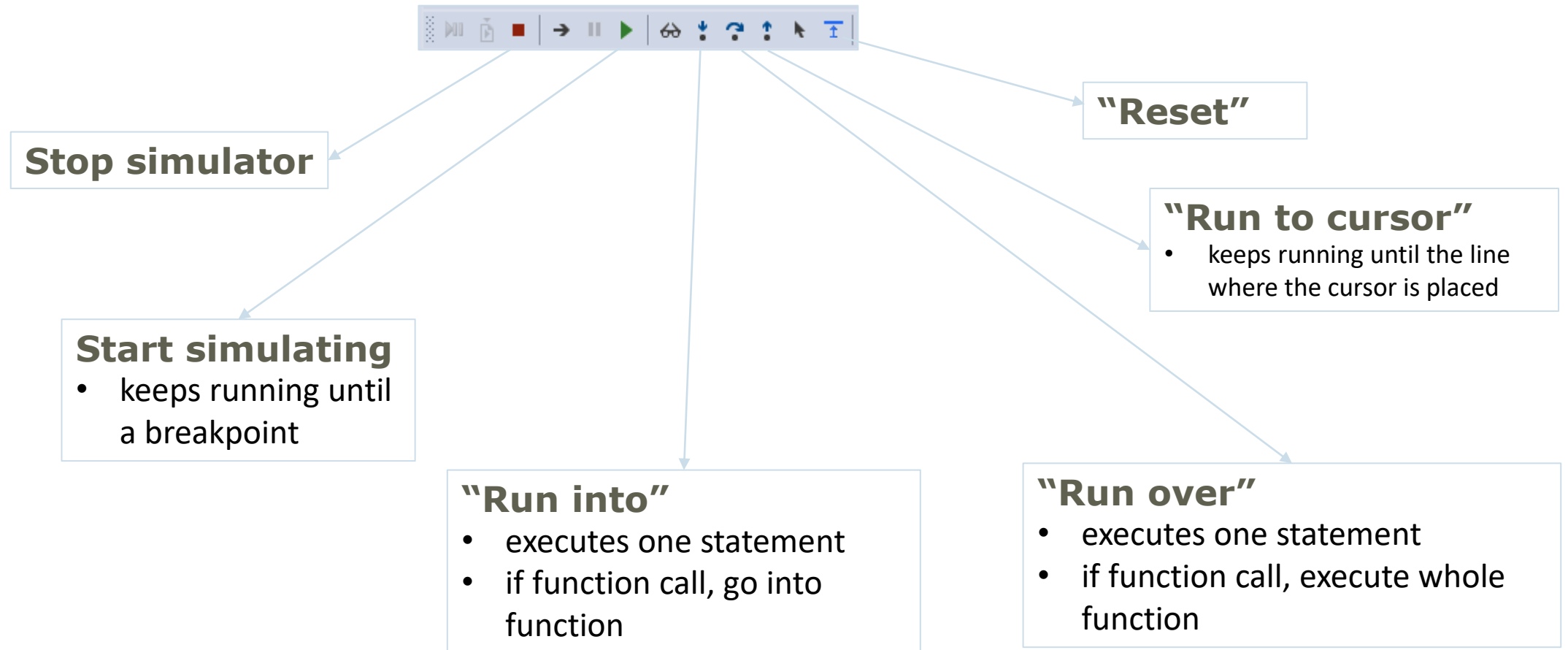
; BOOT code
.org 0x0000
rjmp init

; INIT code
init:
    CBI DDRB,2    ;set PB2 (JOYSTICK) as input
    SBI PORTB,2   ;enable pull-up
    SBI DDRC, 2   ;set PC2 (LED1) as output
    SBI PORTC,2   ;write a logic "1" -> LED OFF

    rjmp main     ;jump to endless loop

; MAIN code
main:
    IN R0,PINB    ;copy PB input to R0
    BST R0,2      ;copy PB2 into T
    BRTC PB2pressed ;jump if JOYSTICK is pressed
    SBI PORTC, 2   ;LED1 OFF
    rjmp main
PB2pressed:
    CBI PORTC, 2   ;LED1 ON
    rjmp main
```

Debug toolbar



Processor status

- Program counter
 - Stack pointer
 - X, Y, and Z register
 - Status register (SREG)
 - I: interrupt flag
 - T: T-flag
 - N: zero flag
 - S: sign flag
 - C: carry flag
- Cycle counter
- Clock frequency
- Stop watch [μ seconds]
- 32 registers, including copies of the X, Y, and Z registers

Processor Status	
Name	Value
Program Counter	0x00000000
Stack Pointer	0x08FF
X Register	0x0000
Y Register	0x0000
Z Register	0x0000
Status Register	I T H S V N Z C
Cycle Counter	0
Frequency	16,000 MHz
Stop Watch	0,00 μ s
Registers	
R00	0x00
R01	0x00
R02	0x00
R03	0x00
R04	0x00
R05	0x00
R06	0x00
R07	0x00
R08	0x00
R09	0x00
R10	0x00
R11	0x00
R12	0x00
R13	0x00
R14	0x00
R15	0x00
R16	0x00
R17	0x00
R18	0x00
R19	0x00
R20	0x00
R21	0x00
R22	0x00
R23	0x00
R24	0x00
R25	0x00
R26	0x00

Input/Output window

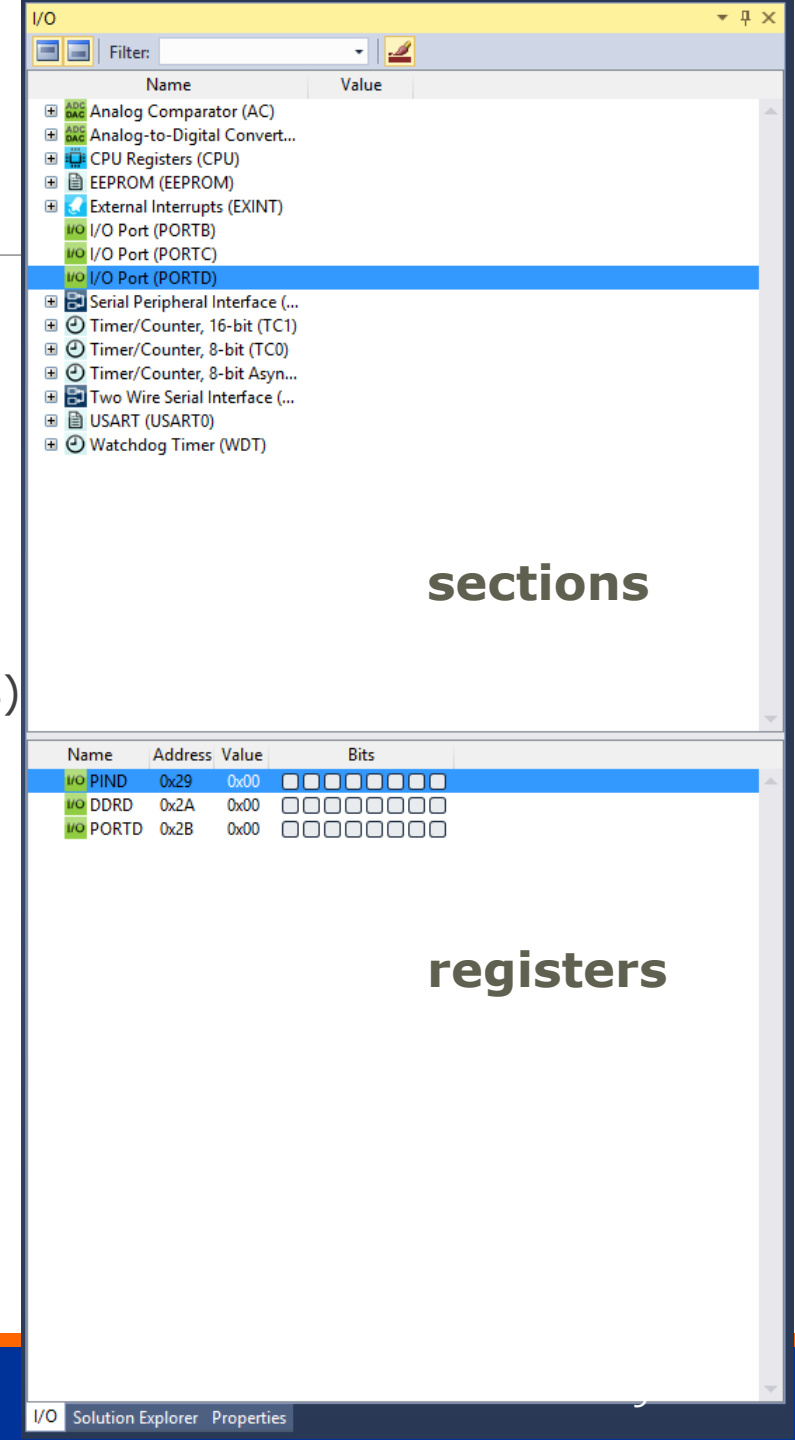
List of functional units

Check settings of timers, ADC, I/O ports, ...

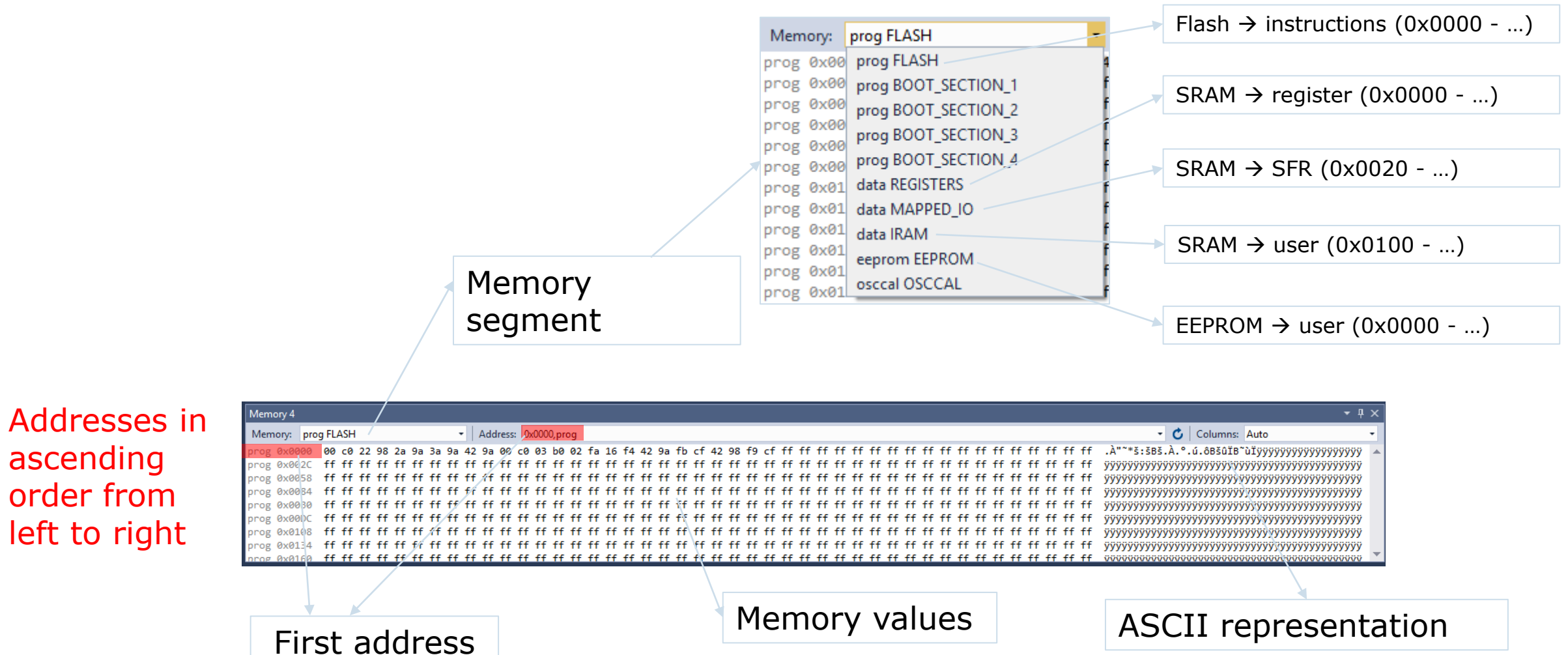
Unfold sections to view functional units registers

Check their values

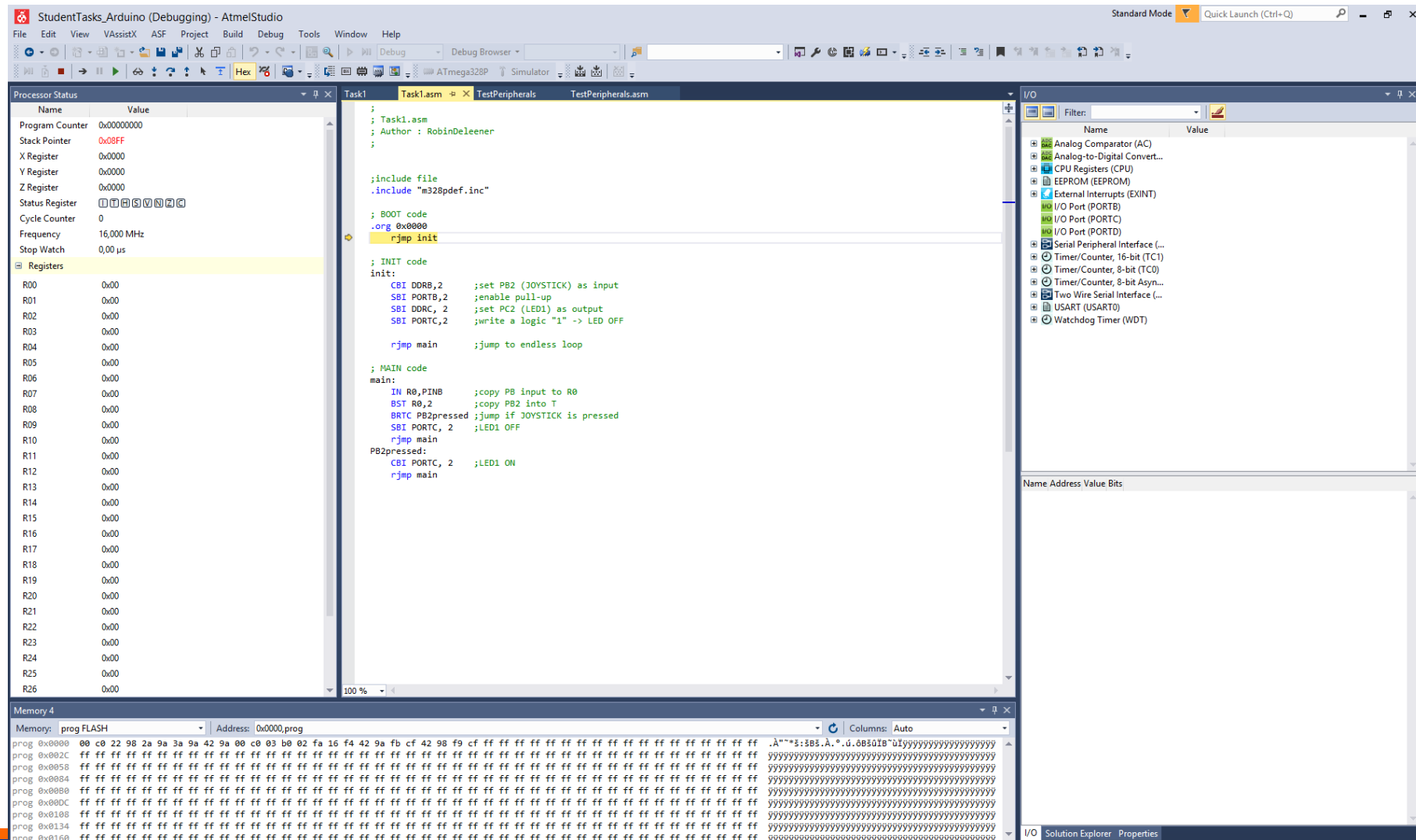
Change their values for simulating external pin changes (PINx registers)

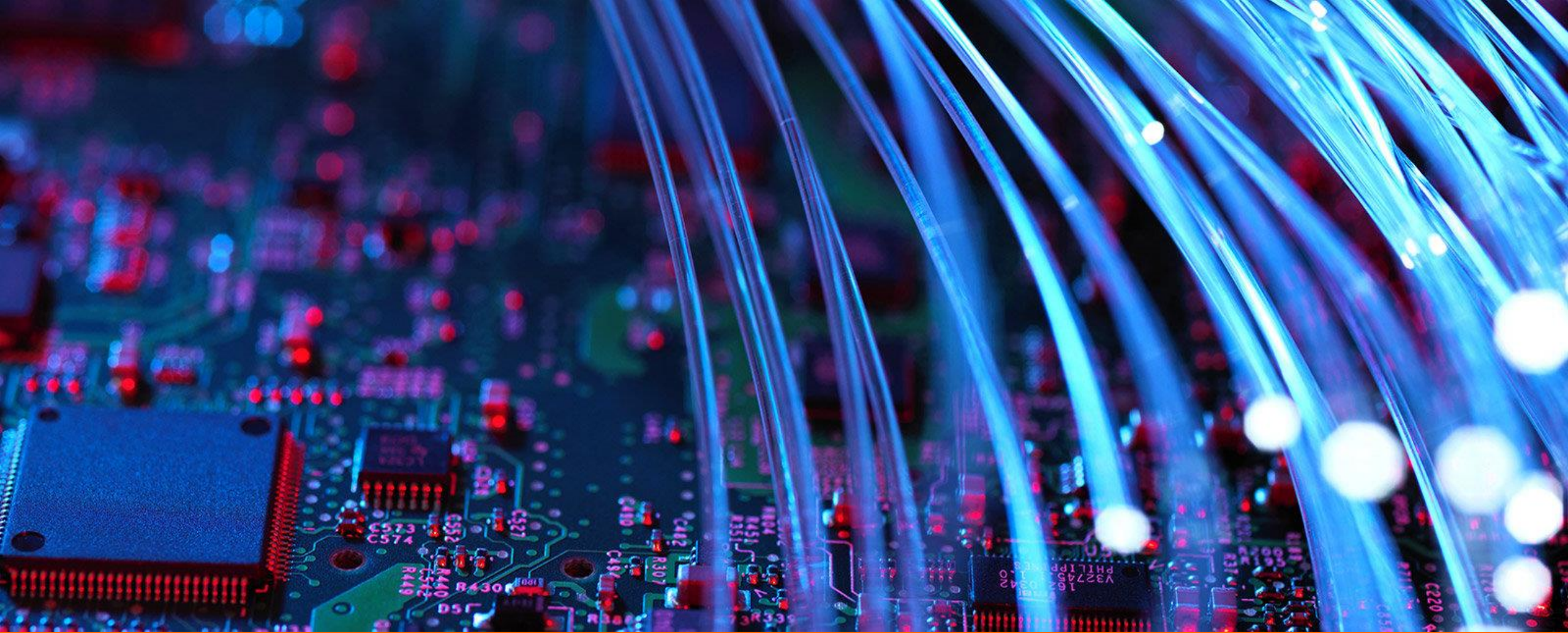


Memory overview



Atmel studio simulator





End of Part 3: simulator