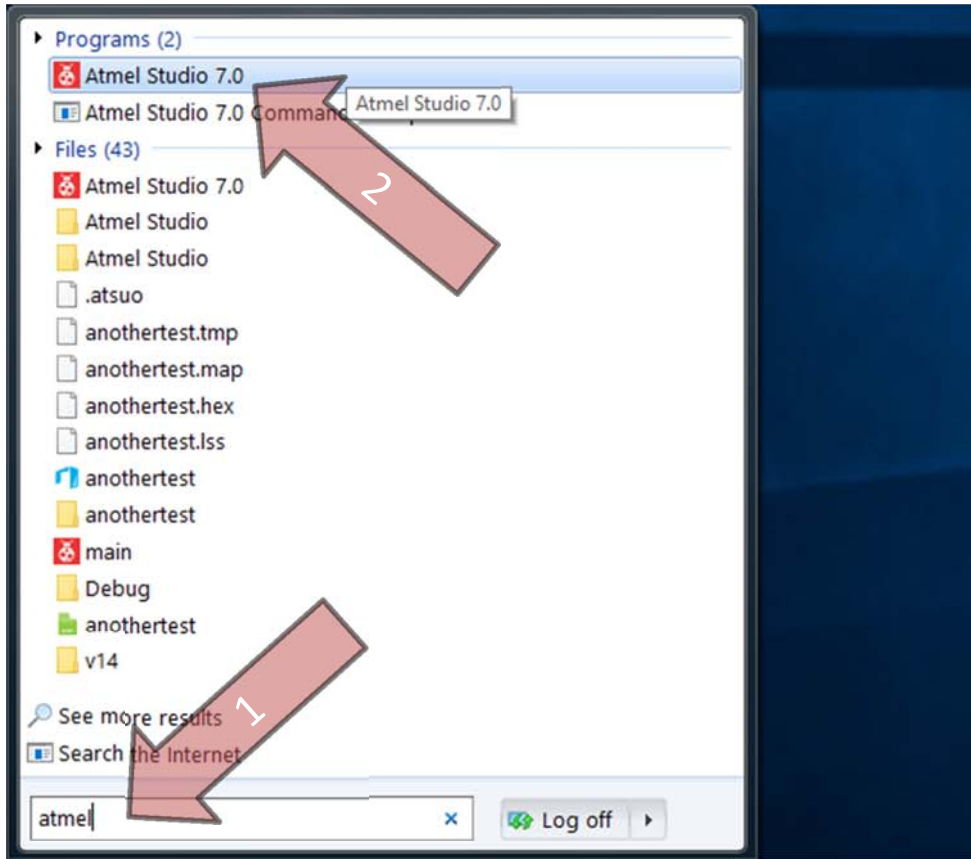


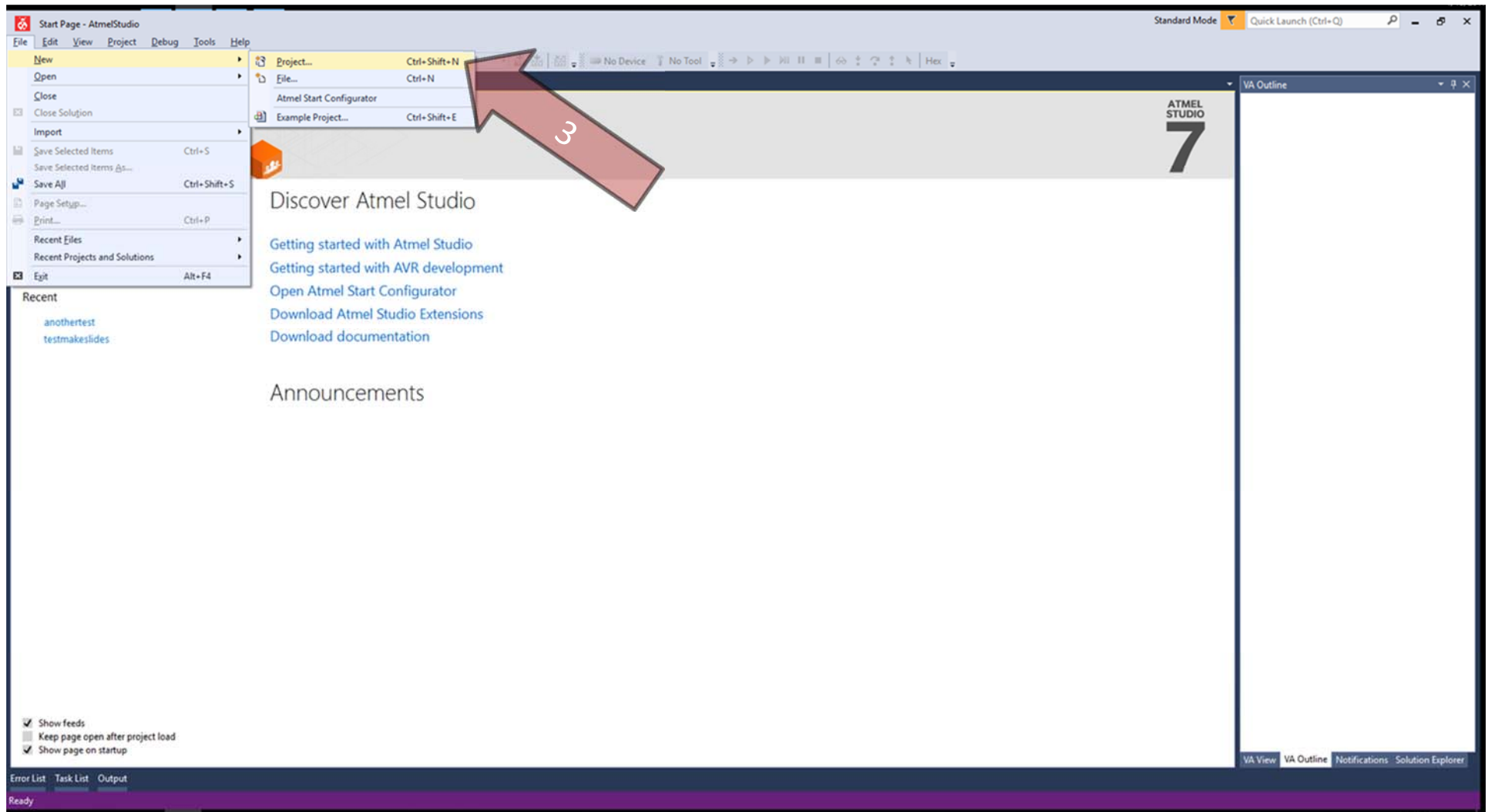
Building Projects and Device Programming using Atmel Studio 7

V5.0

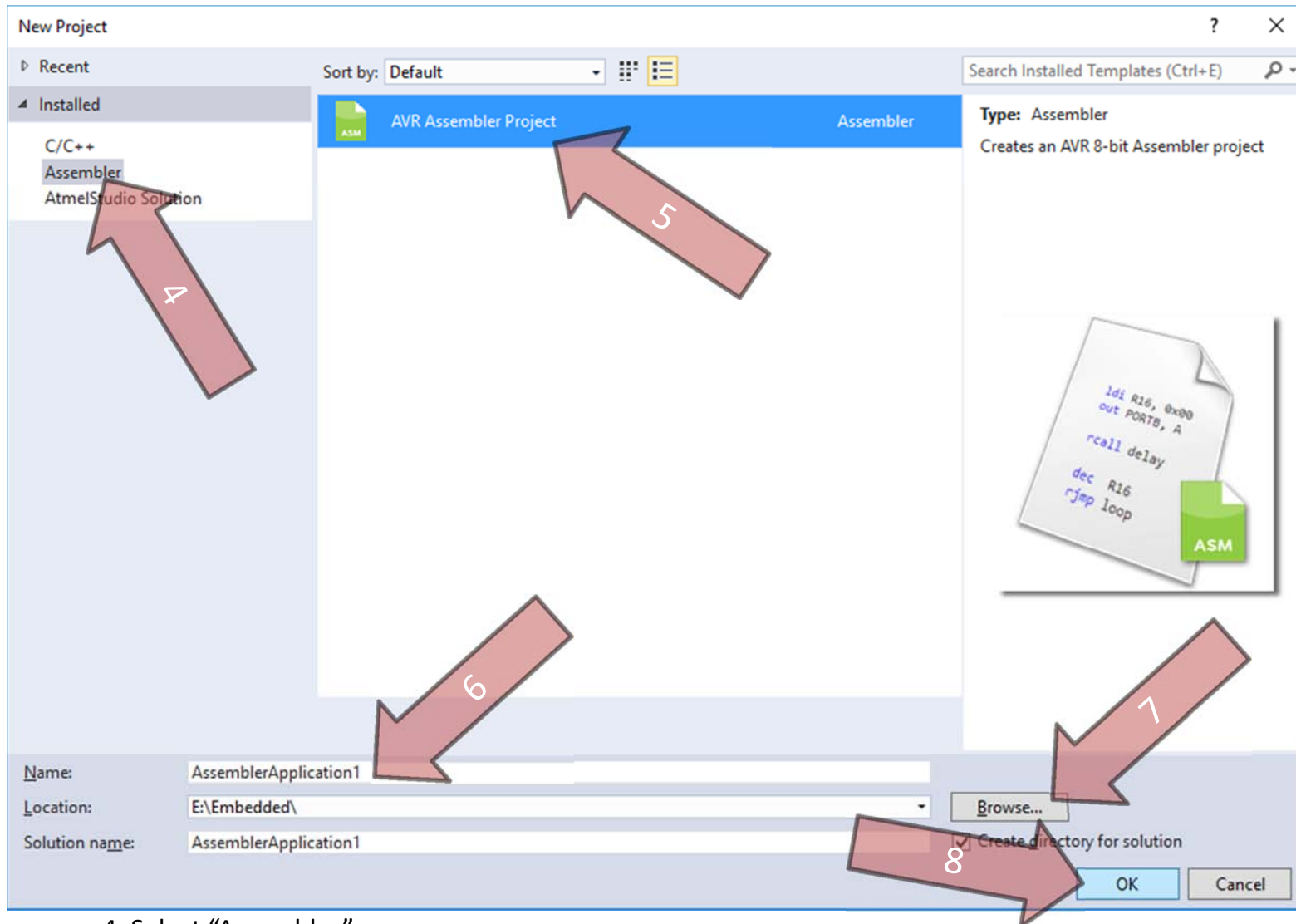
Embedded Systems



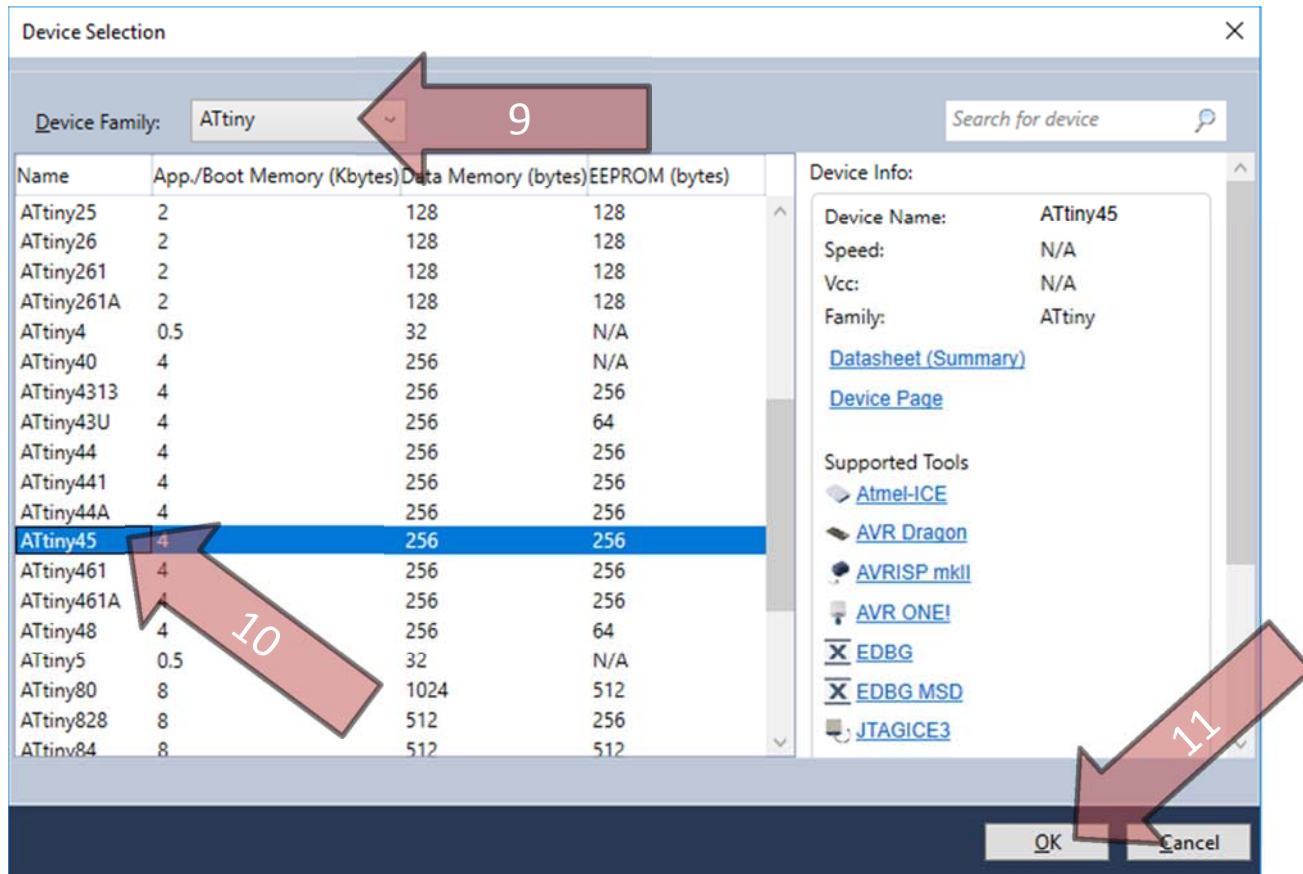
1. Search for Atmel in the start menu
2. Locate Atmel Studio 7.0 and open it



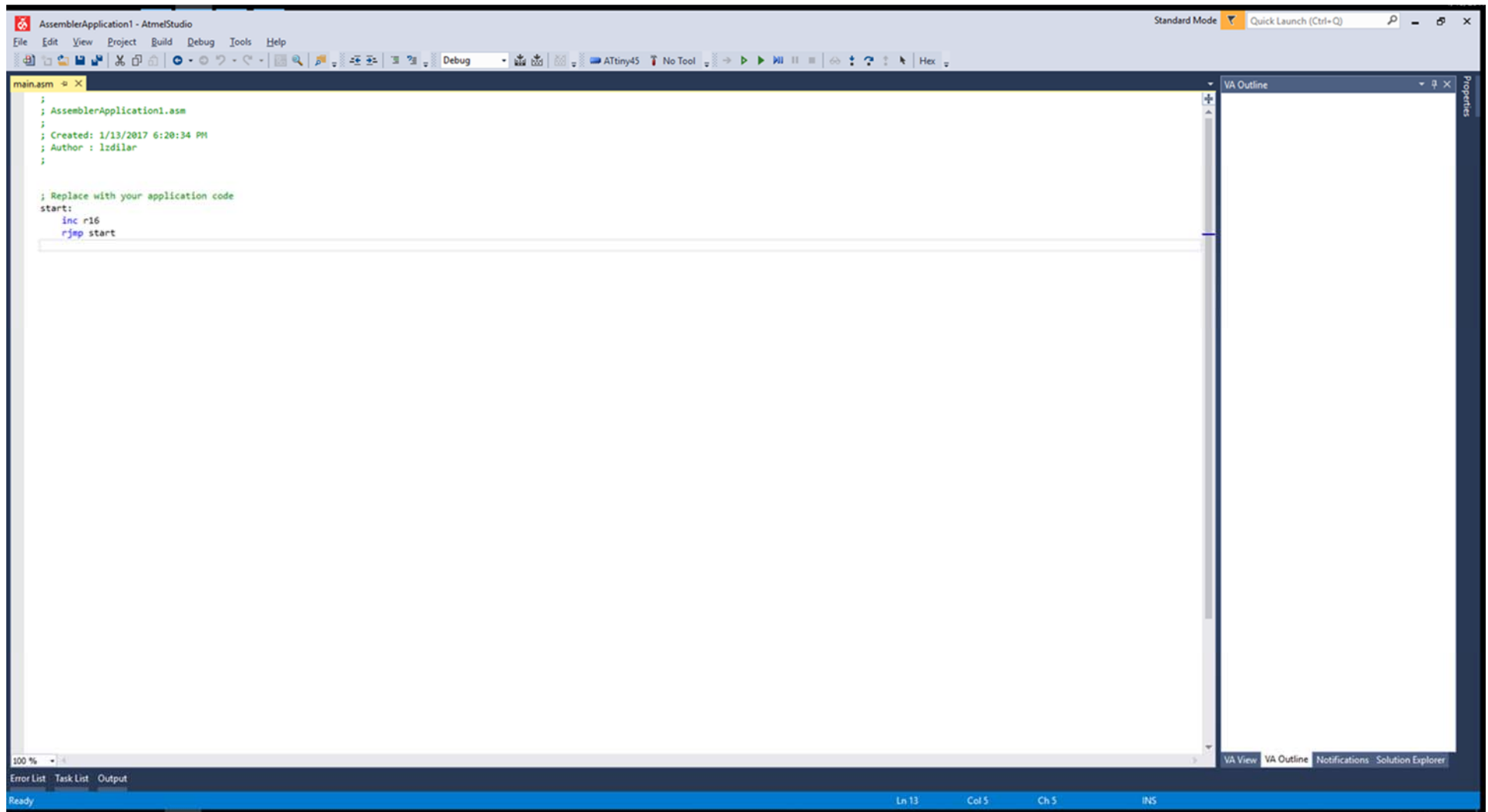
3. When the program opens, navigate to “File” -> “New” -> “Project...” and click it



4. Select "Assembler"
5. Select "AVR Assembler Project"
6. Name the Project
7. Select the save location
8. Click "OK"



9. Select "ATtiny" from the drop down
10. Select "ATtiny45" from the list
11. Click "OK"



The main.asm file should appear with some default code

user.engineering.uiowa.edu/~rbeichel/lectures/es_s17/lab1/index.html

Embedded Systems - ECE:3360

Spring 2017

THE UNIVERSITY OF IOWA


main labs resources policies

Lab 1 Assignment

General
Objective: The objective of this lab is to assemble, test, and debug the ATtiny45 development board (Kit A) that you will use during the first few labs.
Dates: **January 17 – January 27.**
No pre-lab report required.
Check off due no later than **Friday, January 27** (see check off sign up sheet for details!)

Pre-Lab Activity

1. Notify us about your lab partner choice or of your desire to have us assign a lab partner for you; you can find a team sign up sheet in the ECE office (4016 SC).
Deadline: January 21.
2. Purchase the Kit A from the Engineering Electronics Shop (2018 SC). Only one kit is needed per lab team.
3. Optional: sign-up with our lab partner for an [introductory session](#) on PC Board assembly and intro to Atmel (AVR) Studio 6. The sign up sheet can be found in the ECE office (4016 SC). The time slots are: Thursday, Jan 19: 2:30P-4:00P and Friday, Jan 20: 3:00P-4:30P.
4. Take a look at this YouTube [video on electronic soldering](#).



Kit A

Resources
Starter assembly language program, [blinky_v2_lab01.txt](#)
Kit 1 hardware [documentation](#) and assembly [instructions](#).
Introduction to [building projects and programming](#) the ATtiny45 board.
Introduction to [program simulation in Atmel Studio](#).

Lab Activity

- During the formal lab session, students will build/assemble and test the development board. Students will familiarize themselves with the development tools, build a simple circuit, and download a small AVR assembly language program to their board.
- Following the formal lab, students must work on the following tasks.

12. Navigate to the class website for Lab 1
13. Click on the `blinky.txt`

```
Assembly language file for Lab 1 in 55:036 (Embedded Systems)
Spring 2014, The University of Iowa.

LEDs are connected via a 470 Ohm resistor from PB1, PB2 to Vcc

A. Kruger, R. Beichel

.include "to45def.inc"
.cseg
.org 0

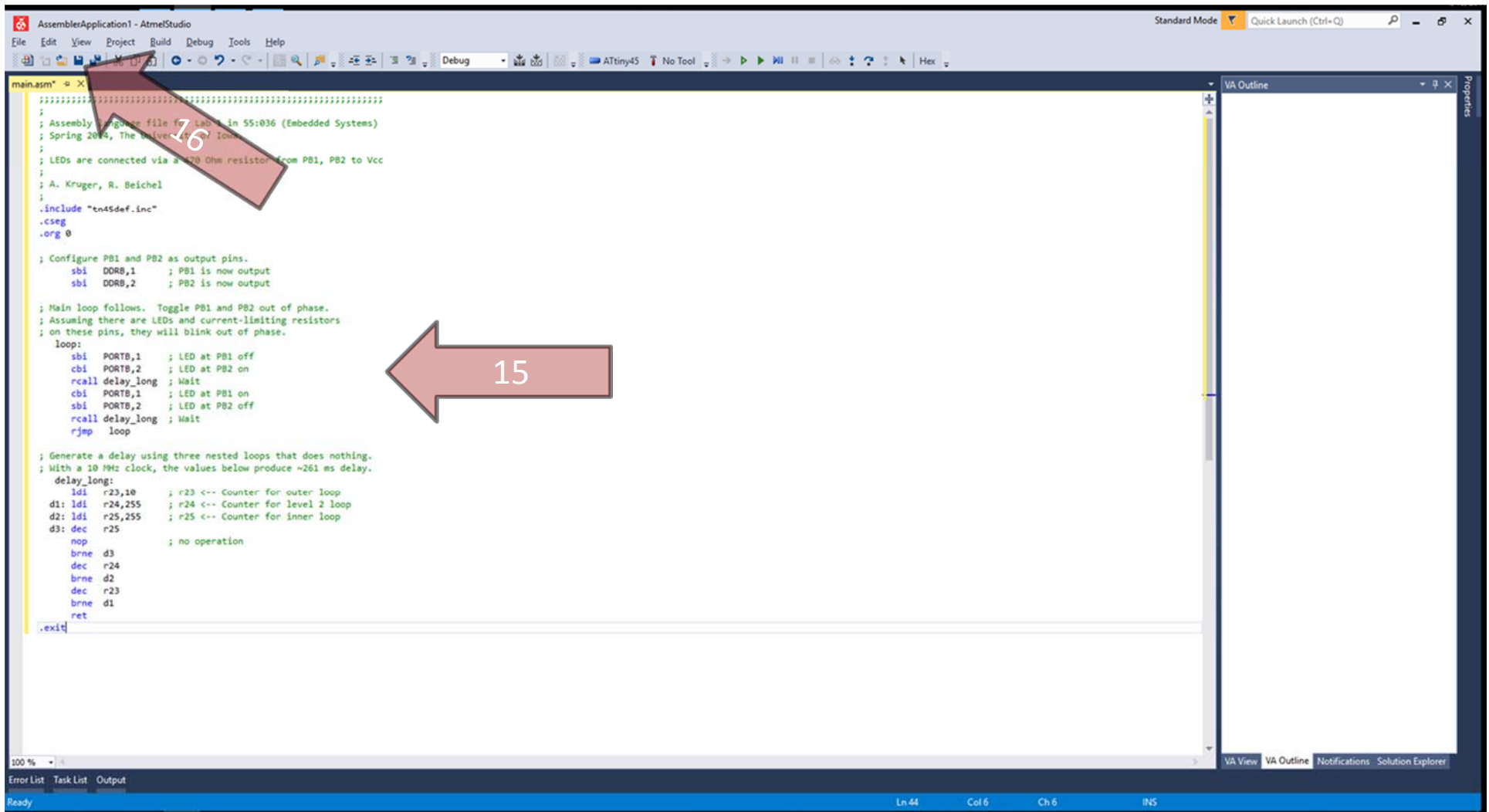
; Configure PB1 and PB2 as output pins.
sbi DDRB,1 ; PB1 is now output
sbi DDRB,2 ; PB2 is now output

; Main loop follows. Toggle PB1 and PB2 out of phase.
; Assuming there are LEDs and current-limiting resistors
; on these pins, they will blink out of phase.
loop:
sbi PORTB,1 ; LED at PB1 off
cbi PORTB,2 ; LED at PB2 on
rcall delay_long ; Wait
cbi PORTB,1 ; LED at PB1 on
sbi PORTB,2 ; LED at PB2 off
rcall delay_long ; Wait
rjmp loop

; Generate a delay using three nested loops that does nothing.
; With a 10 MHz clock, the values below produce ~261 ms delay.
delay_long:
ldi r23,10 ; r23 <-- Counter for outer loop
d1: ldi r24,255 ; r24 <-- Counter for level 2 loop
d2: ldi r25,255 ; r25 <-- Counter for inner loop
d3: dec r25
nop ; no operation
brne d3
dec r24
brne d2
dec r23
brne d1
ret
.exit
```

14

14. Copy all of the code



AssemblerApplication1 - AtmelStudio

File Edit View Project Build Debug Tools Help

Debug ATtiny45 No Tool

main.asm

```

;=====
; Assembly language file for Lab 4 in 55:036 (Embedded Systems)
; Spring 2004, The University of Iowa
;
; LEDs are connected via a 470 Ohm resistor from PB1, PB2 to Vcc
;
; A. Kruger, R. Beichel
;
.include "tn45def.inc"
.cseg
.org 0

; Configure PB1 and PB2 as output pins.
sbi DDRB,1 ; PB1 is now output
sbi DDRB,2 ; PB2 is now output

; Main loop follows. Toggle PB1 and PB2 out of phase.
; Assuming there are LEDs and current-limiting resistors
; on these pins, they will blink out of phase.
loop:
sbi PORTB,1 ; LED at PB1 off
cbi PORTB,2 ; LED at PB2 on
rcall delay_long ; Wait
cbi PORTB,1 ; LED at PB1 on
sbi PORTB,2 ; LED at PB2 off
rcall delay_long ; Wait
rjmp loop

; Generate a delay using three nested loops that does nothing.
; With a 10 MHz clock, the values below produce ~261 ms delay.
delay_long:
ldi r23,10 ; r23 <-- Counter for outer loop
d1: ldi r24,255 ; r24 <-- Counter for level 2 loop
d2: ldi r25,255 ; r25 <-- Counter for inner loop
d3: dec r25
nop ; no operation
brne d3
dec r24
brne d2
dec r23
brne d1
ret
.exit

```

100 %

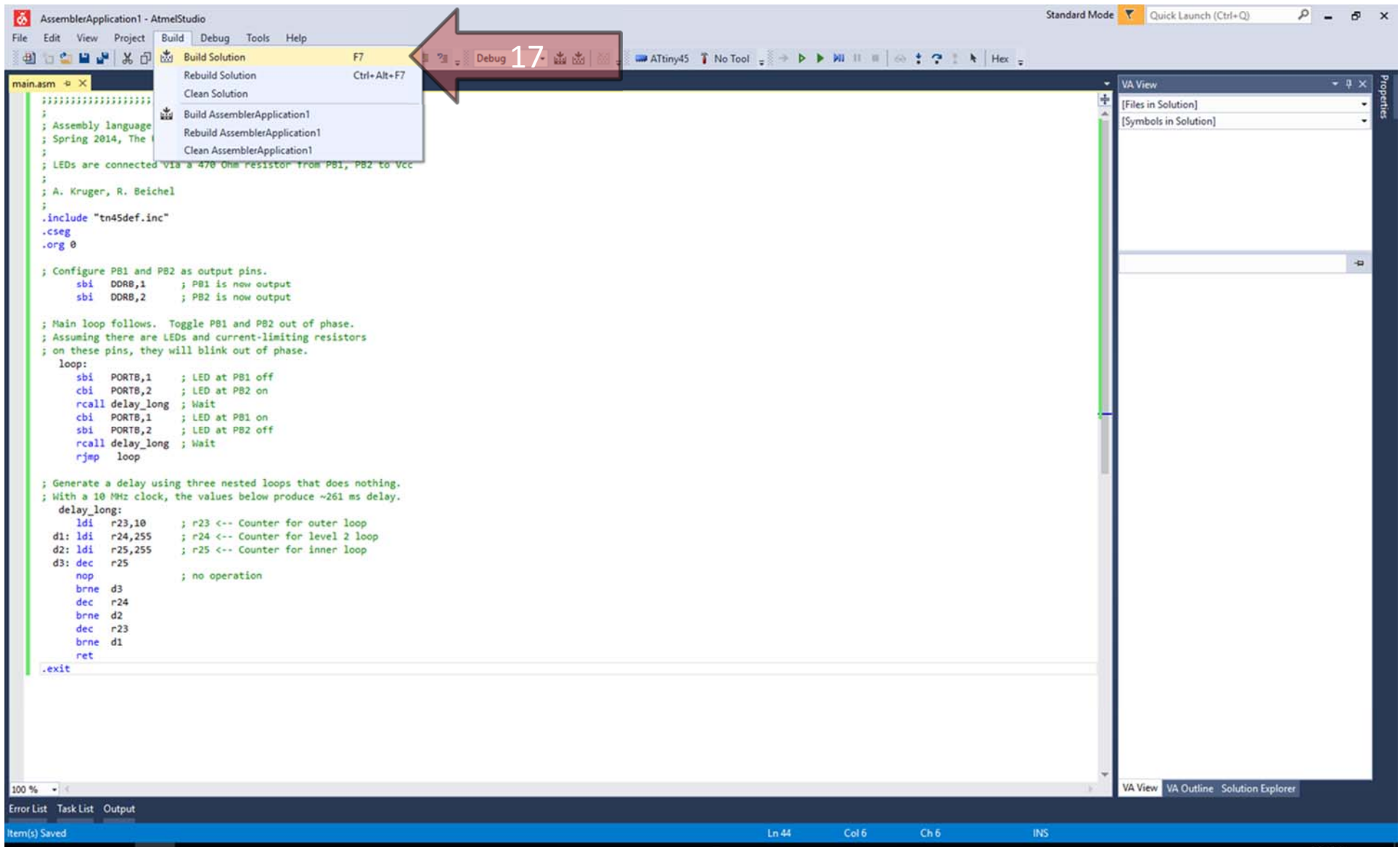
Error List Task List Output

Ready

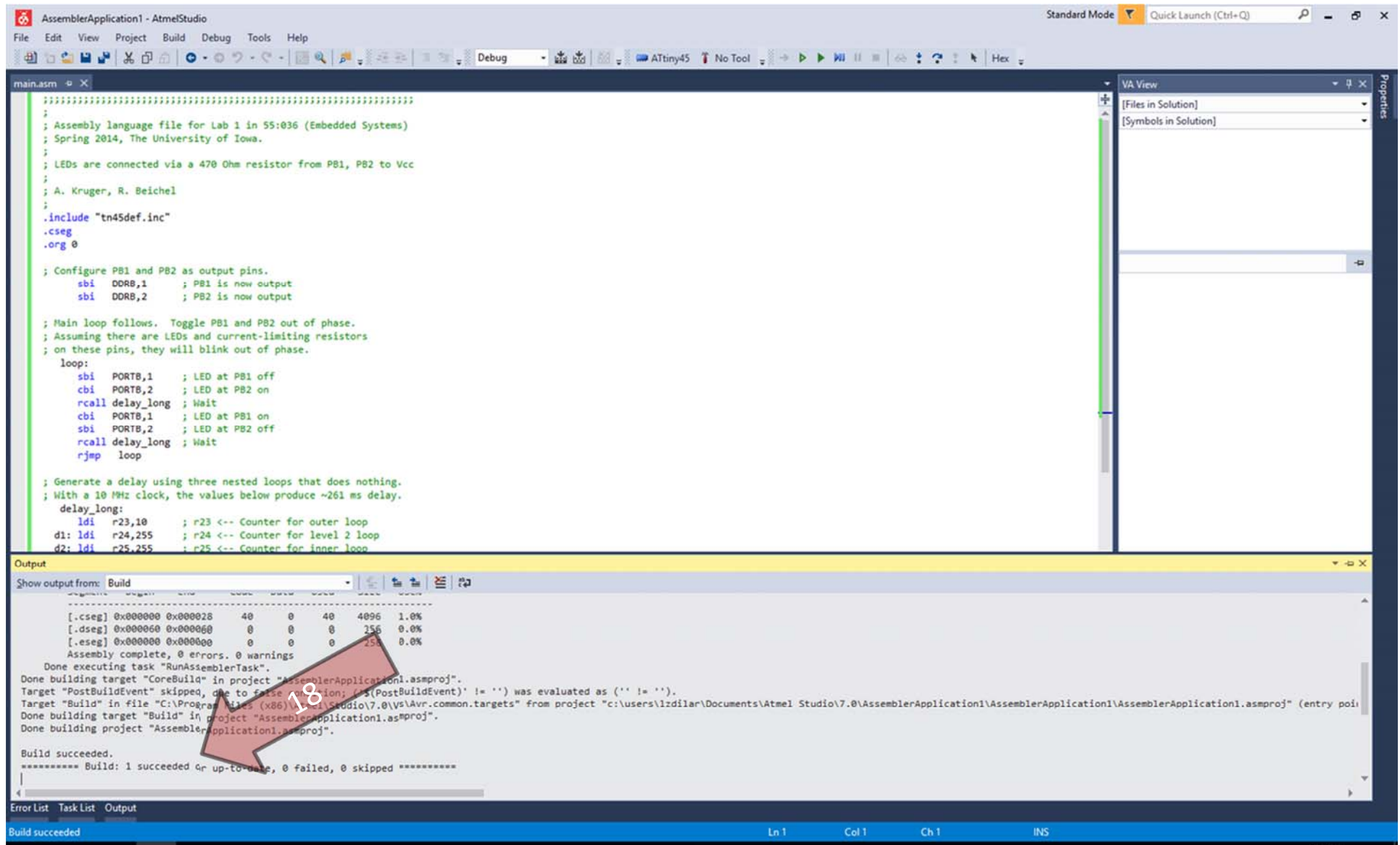
Ln 44 Col 6 Ch 6 INS

15. Paste the code into main.asm

16. Save main.asm

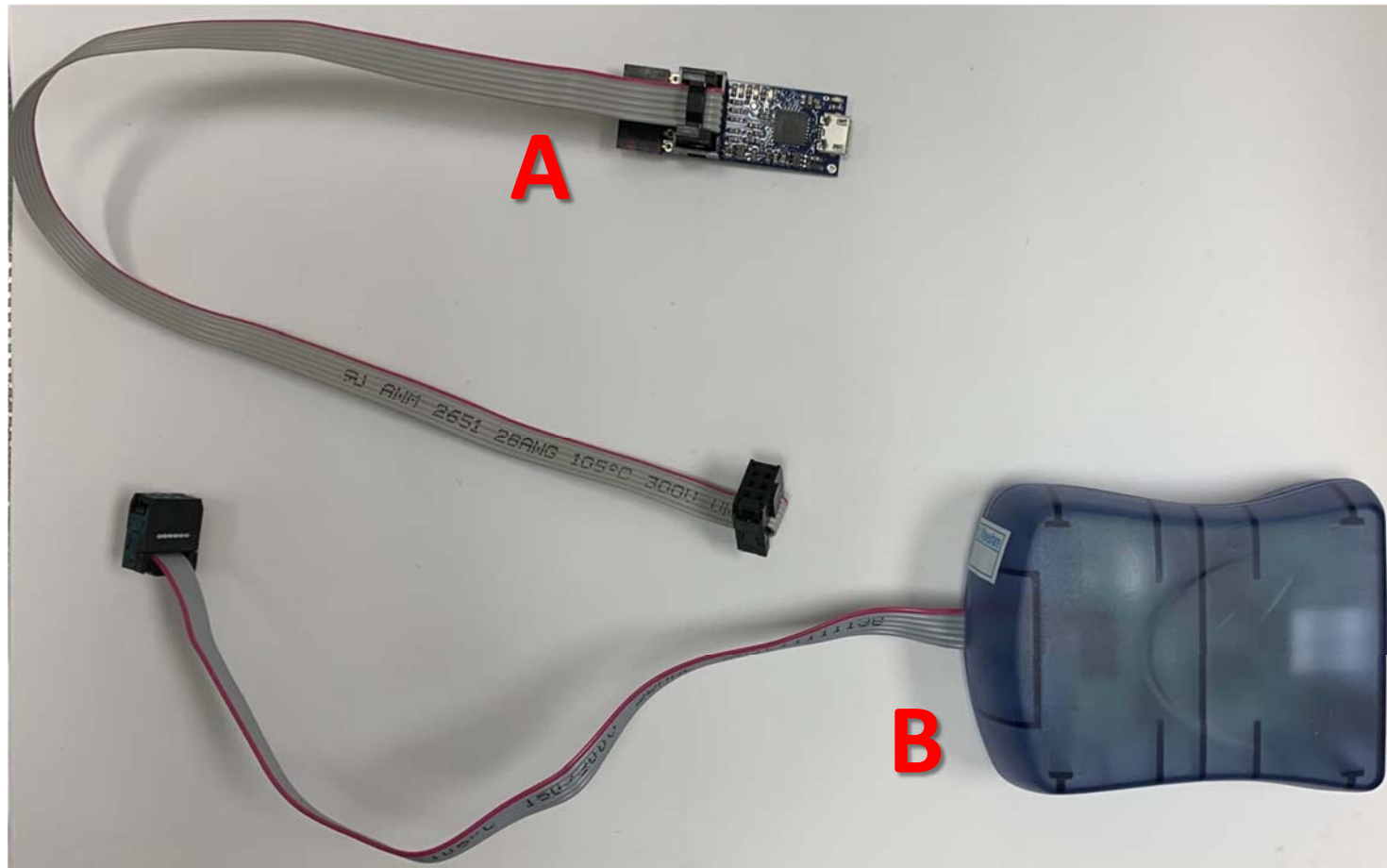


17. Navigate to “Build” -> “Build Solution” and click it



18. Wait for the build to complete

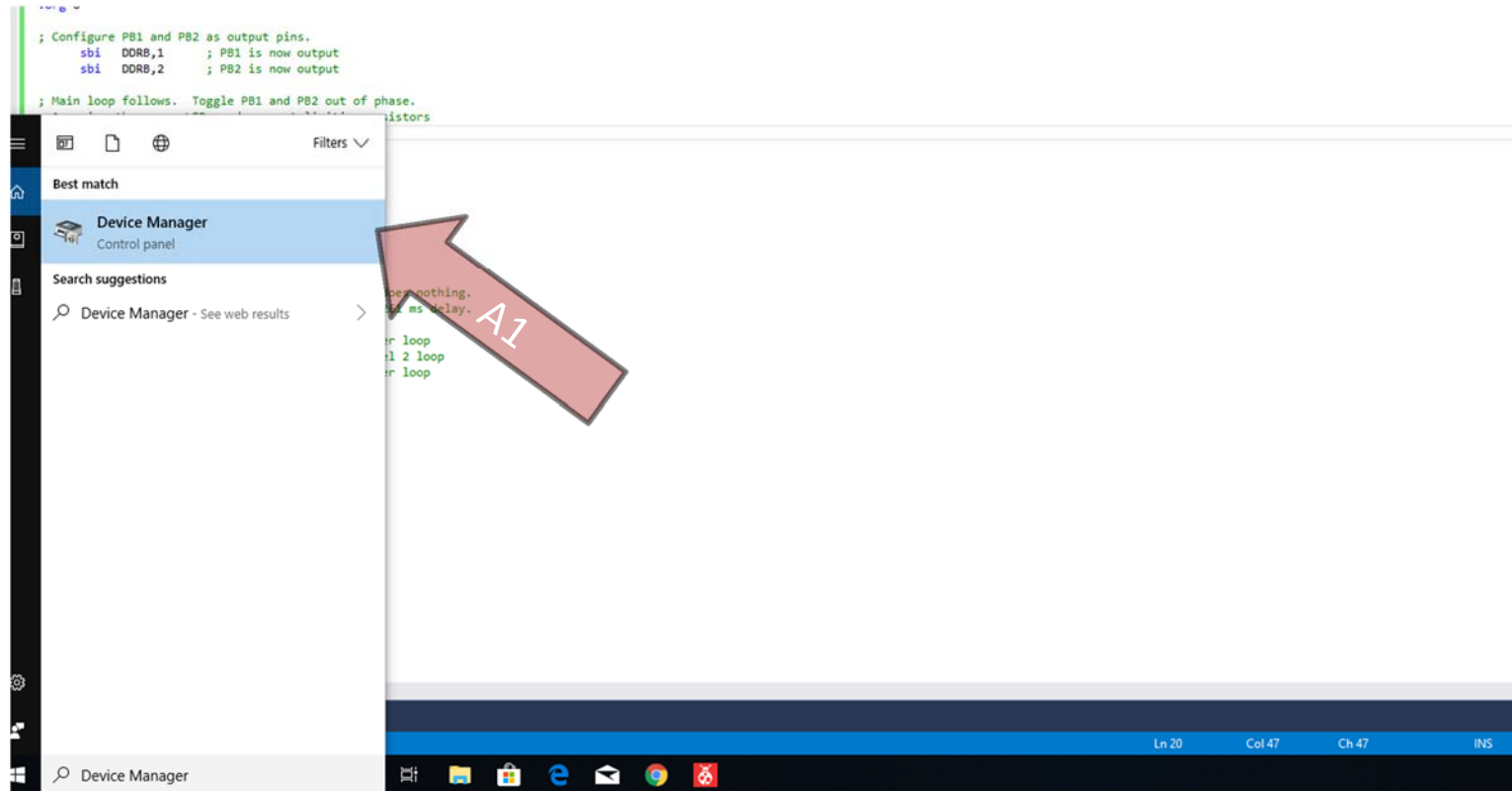
AVR ISP: Two Options



Option A – Pololu ISP: Follow A1-A6, then proceed to Step 19

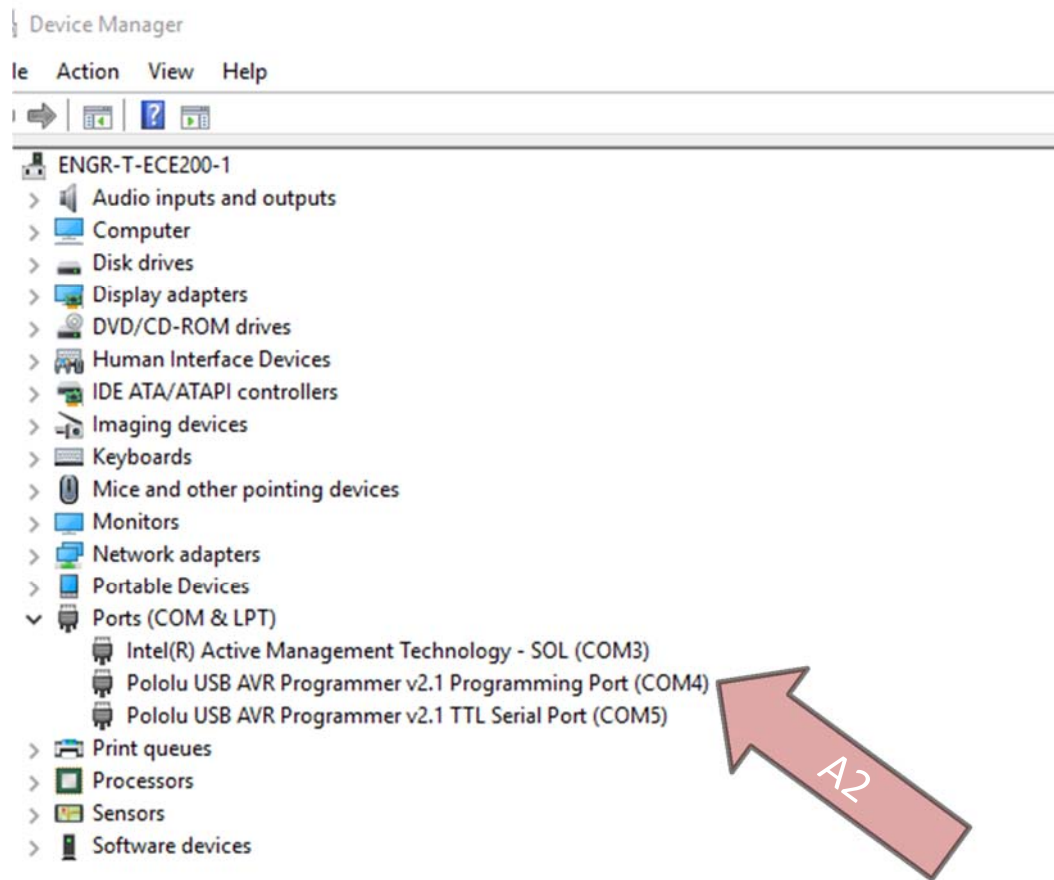
Option B – AVRISP : Go directly to Step 19

Option A: Pololu USB AVR Programmer



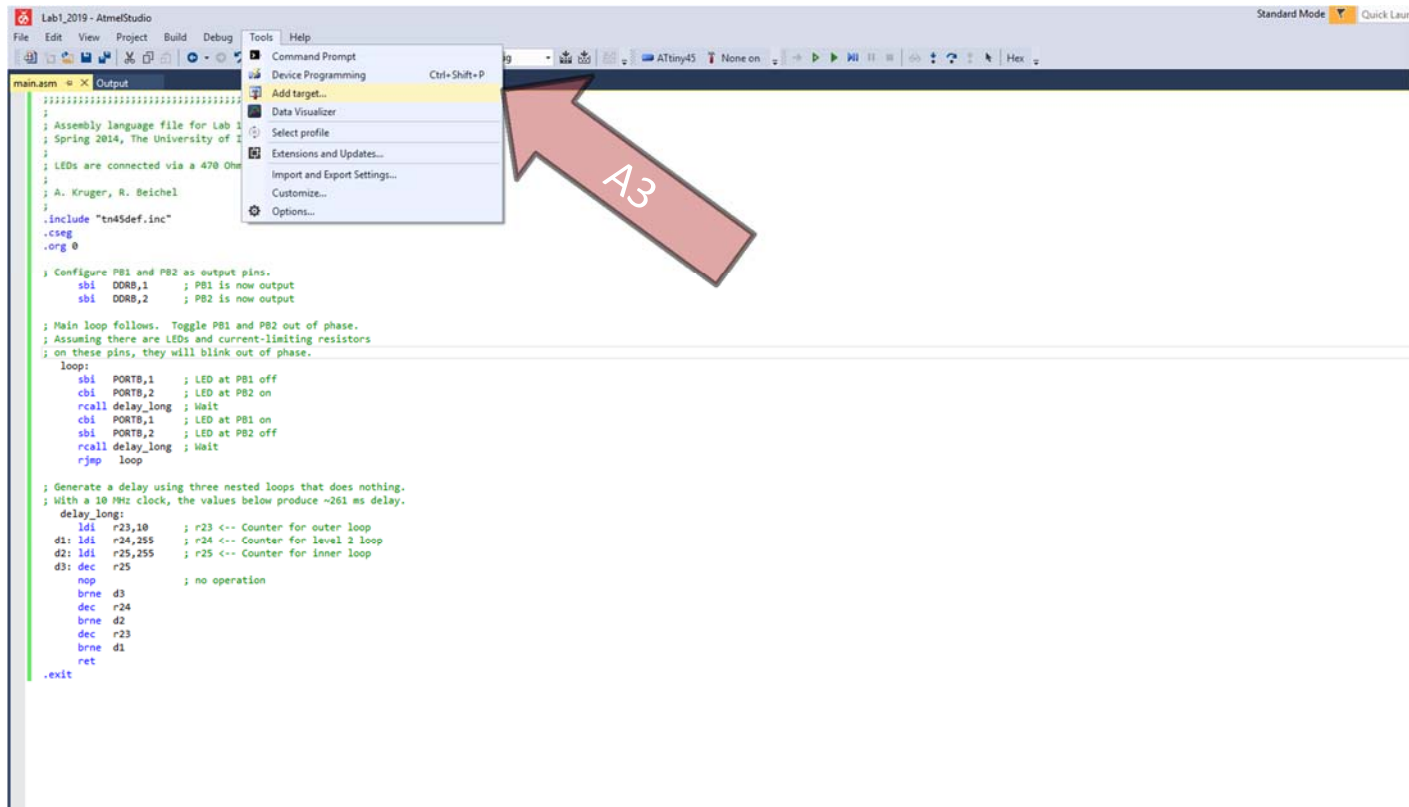
A1) Open "Device Manager" and navigate to "Ports(COM & LPT)"

Option A: Pololu USB AVR Programmer



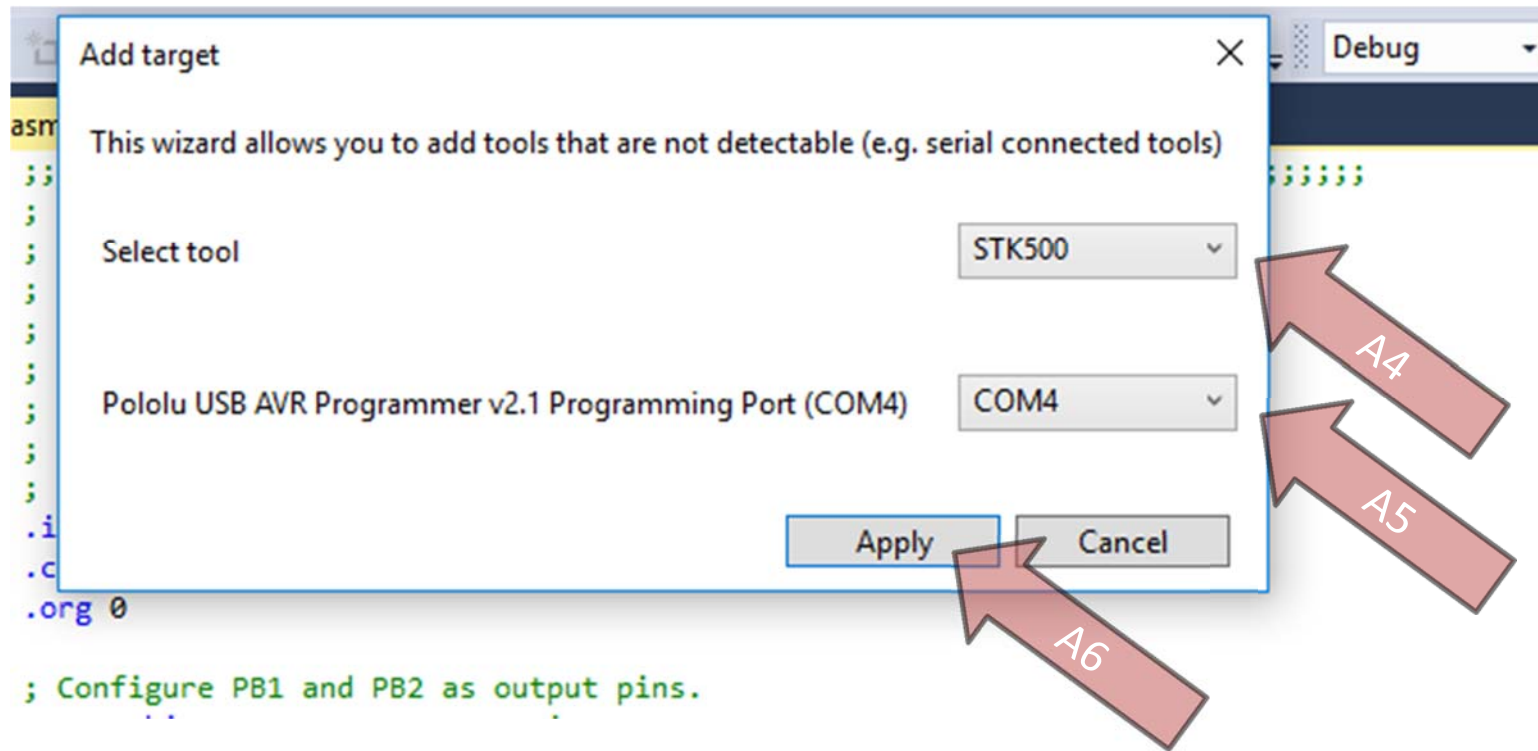
A2) Remember the Programming Port number(COM 4)

Option A: Pololu USB AVR Programmer



A3) Select tools menu and Add Target

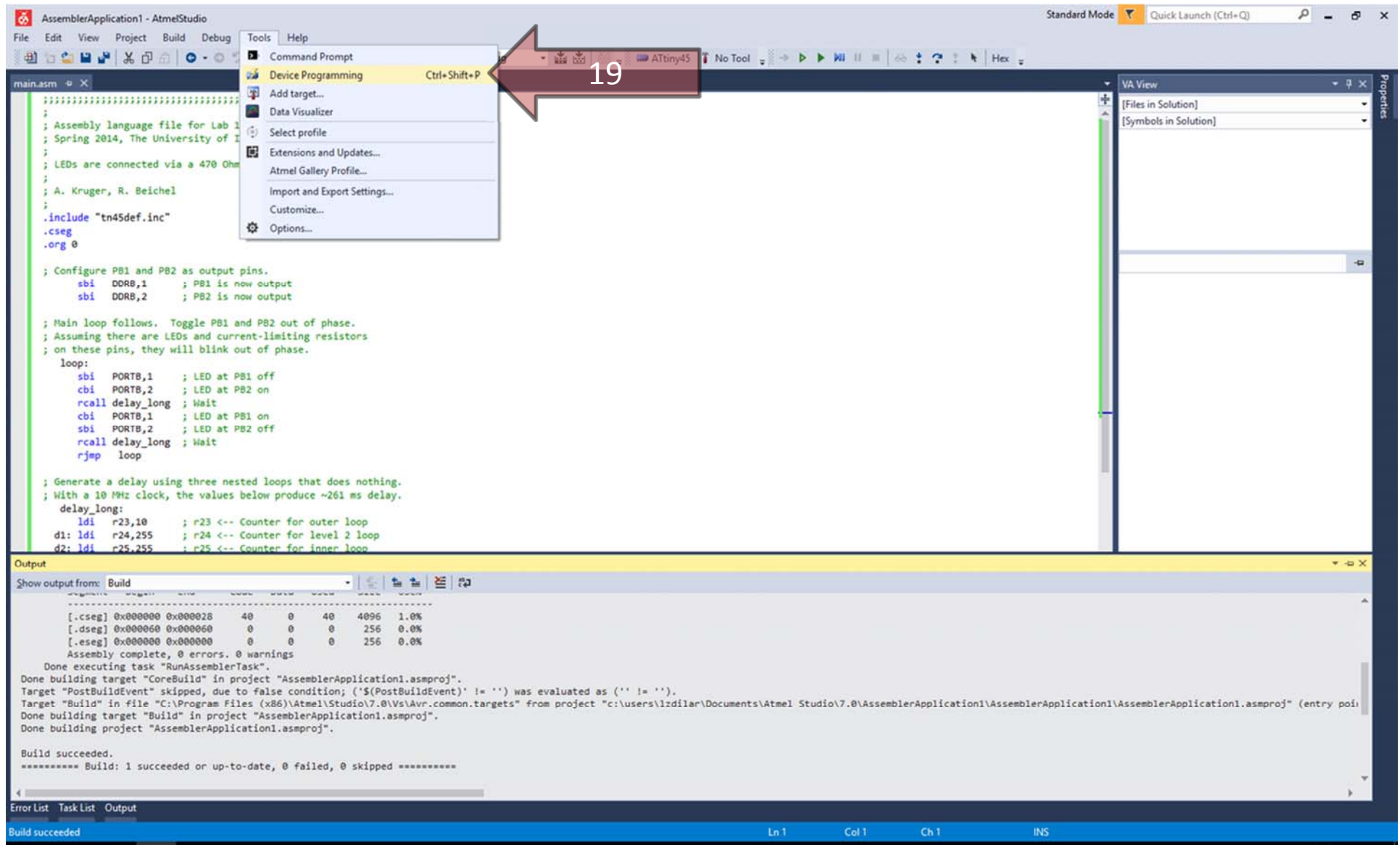
Option A: Pololu USB AVR Programmer



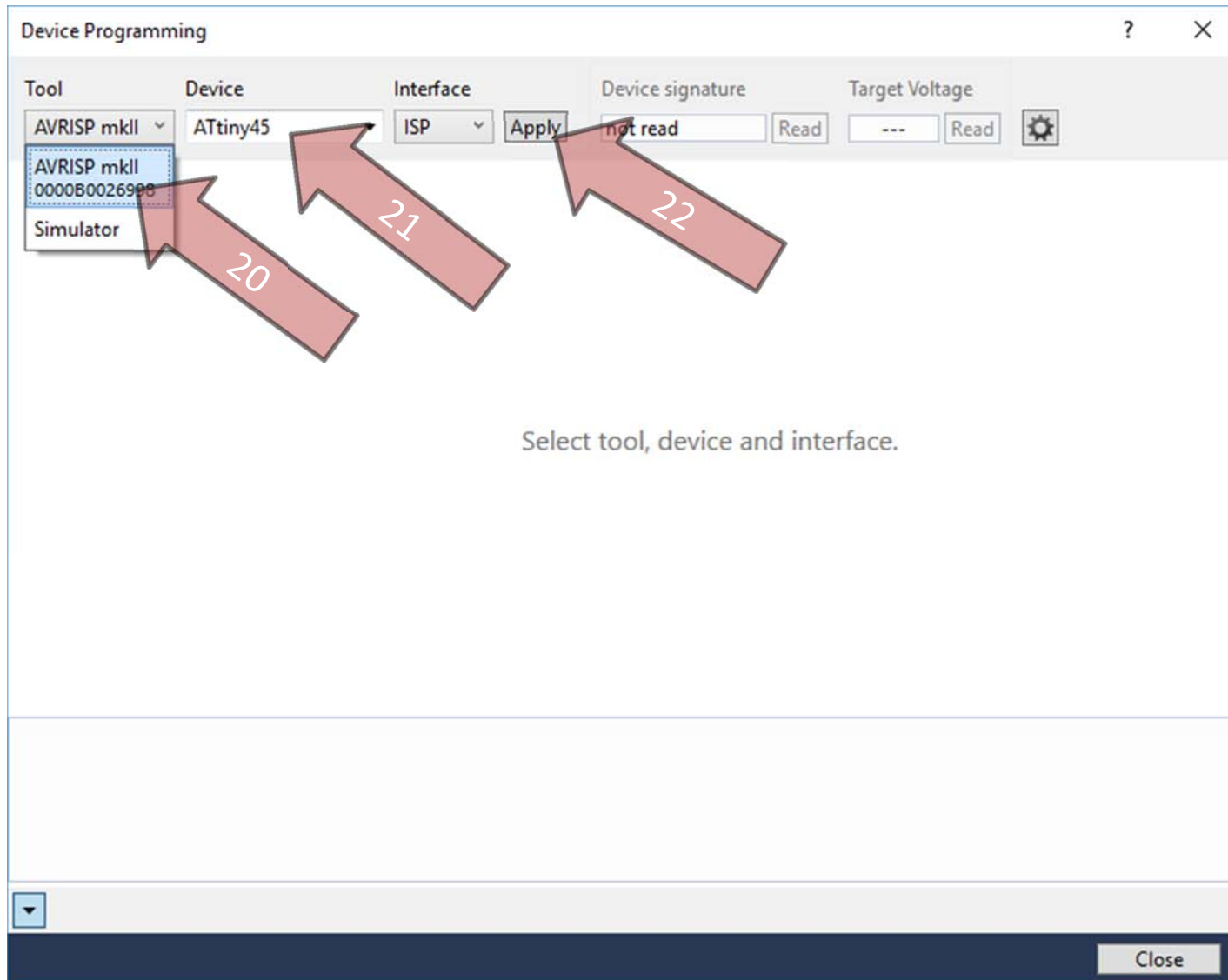
A4) Select STK500 Tool

A5) Select the corresponding COM Port

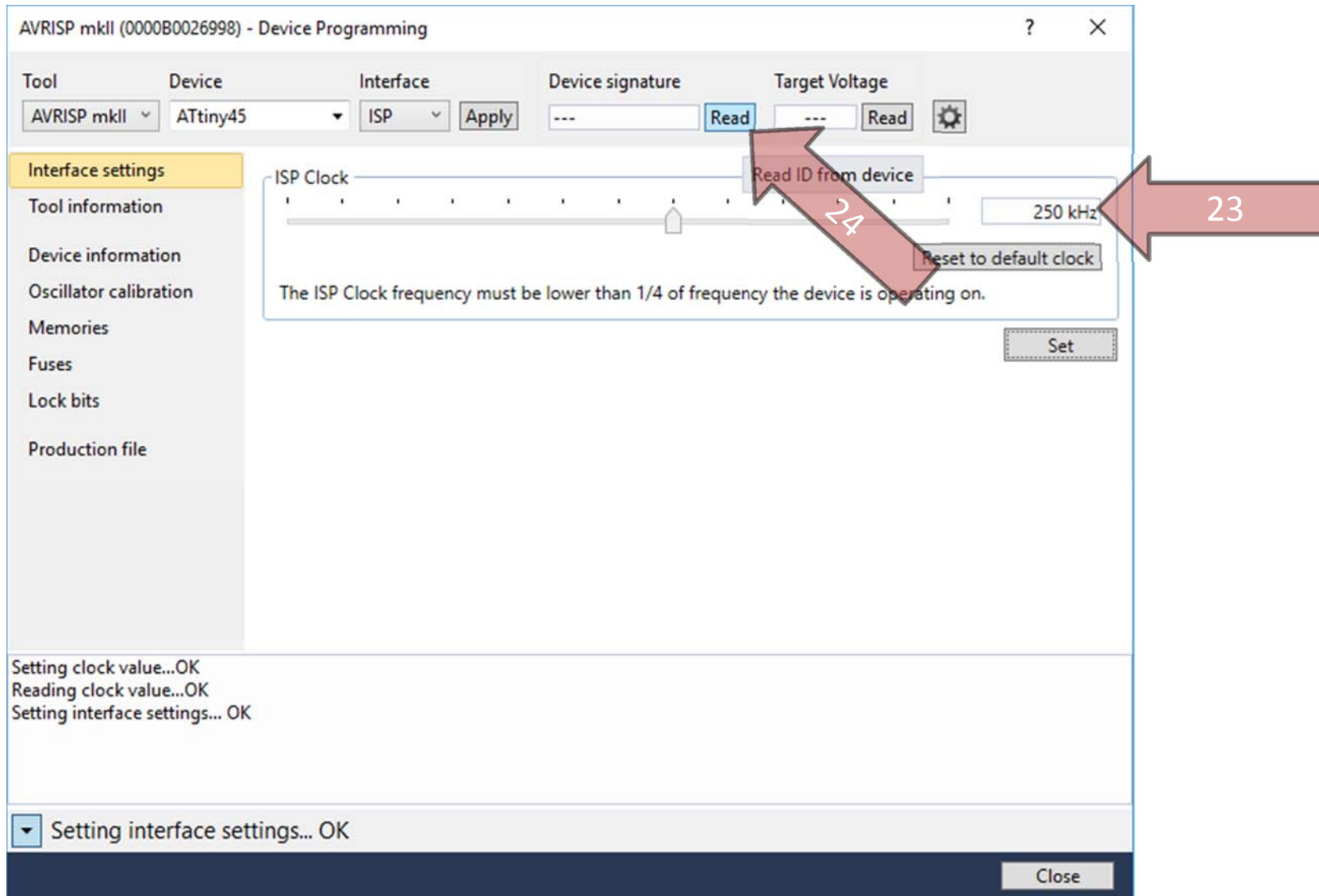
A6) Select Apply



19. Navigate to
“Tools” -> “Device Programming” and click it

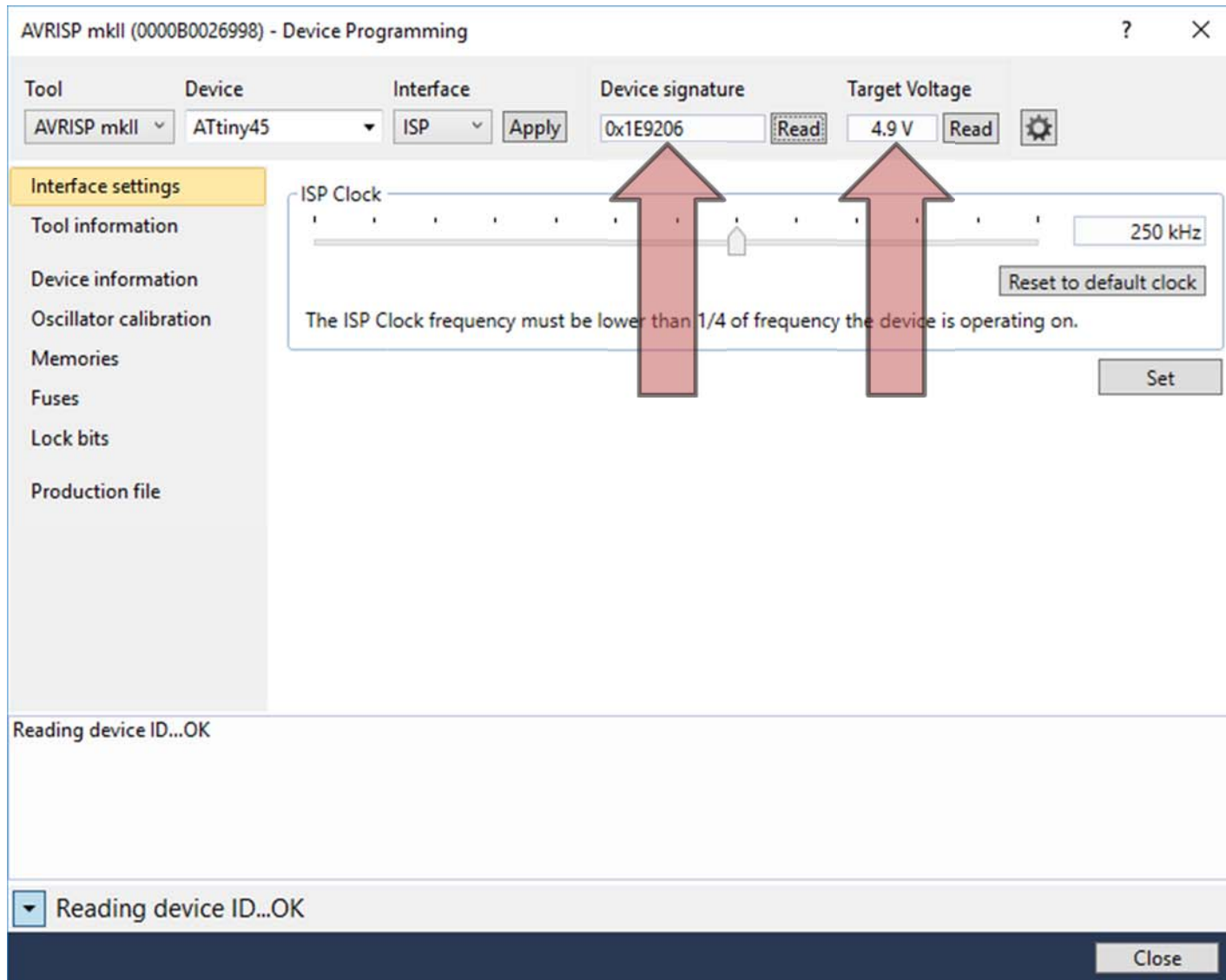


20. **Select (A)"STK500 COM4" (B) "AVRISP" as the "Tool"**
21. Select "ATtiny45" as the "Device"
22. Click "Apply"

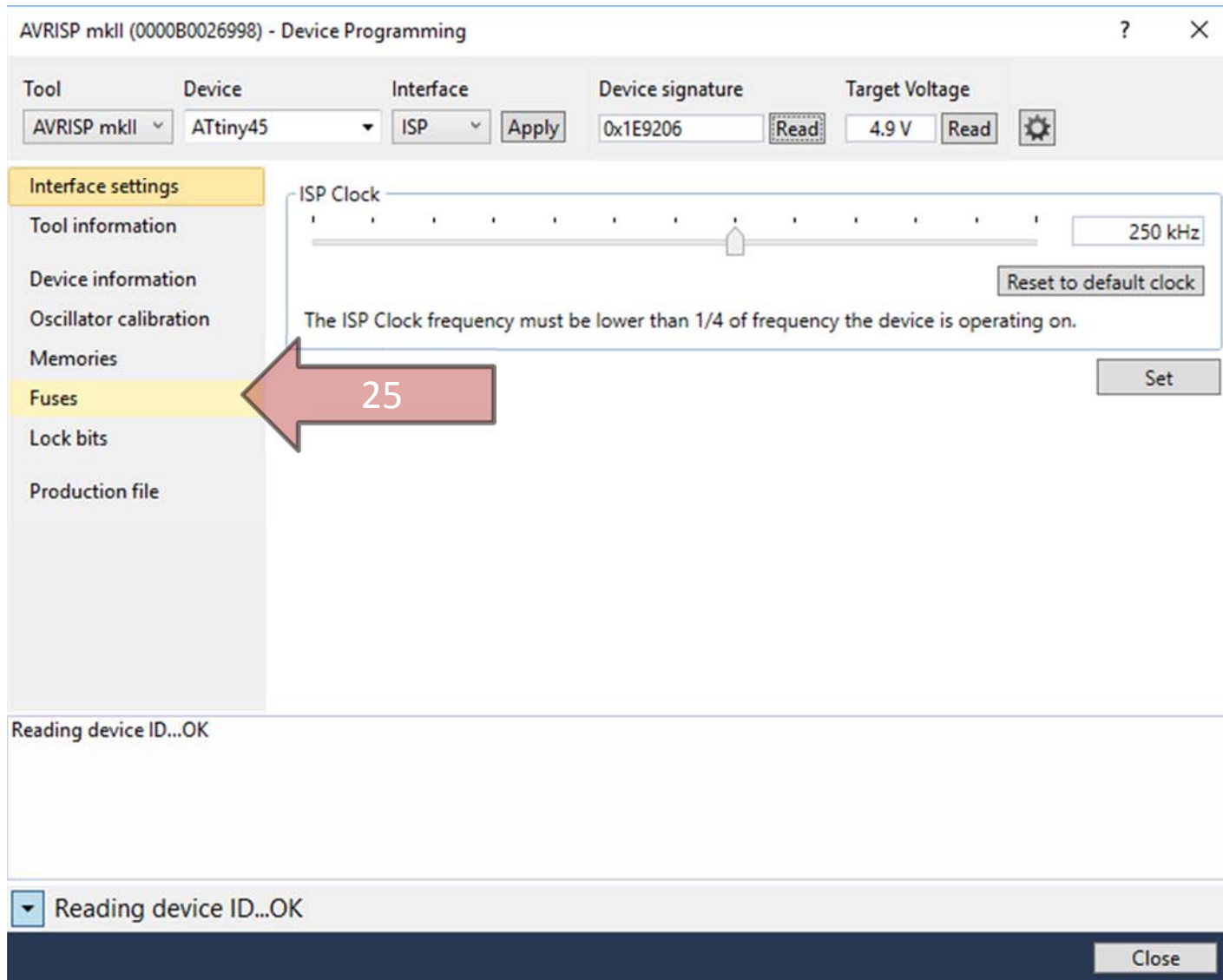


23. Set the ISP clock frequency to 250 kHz

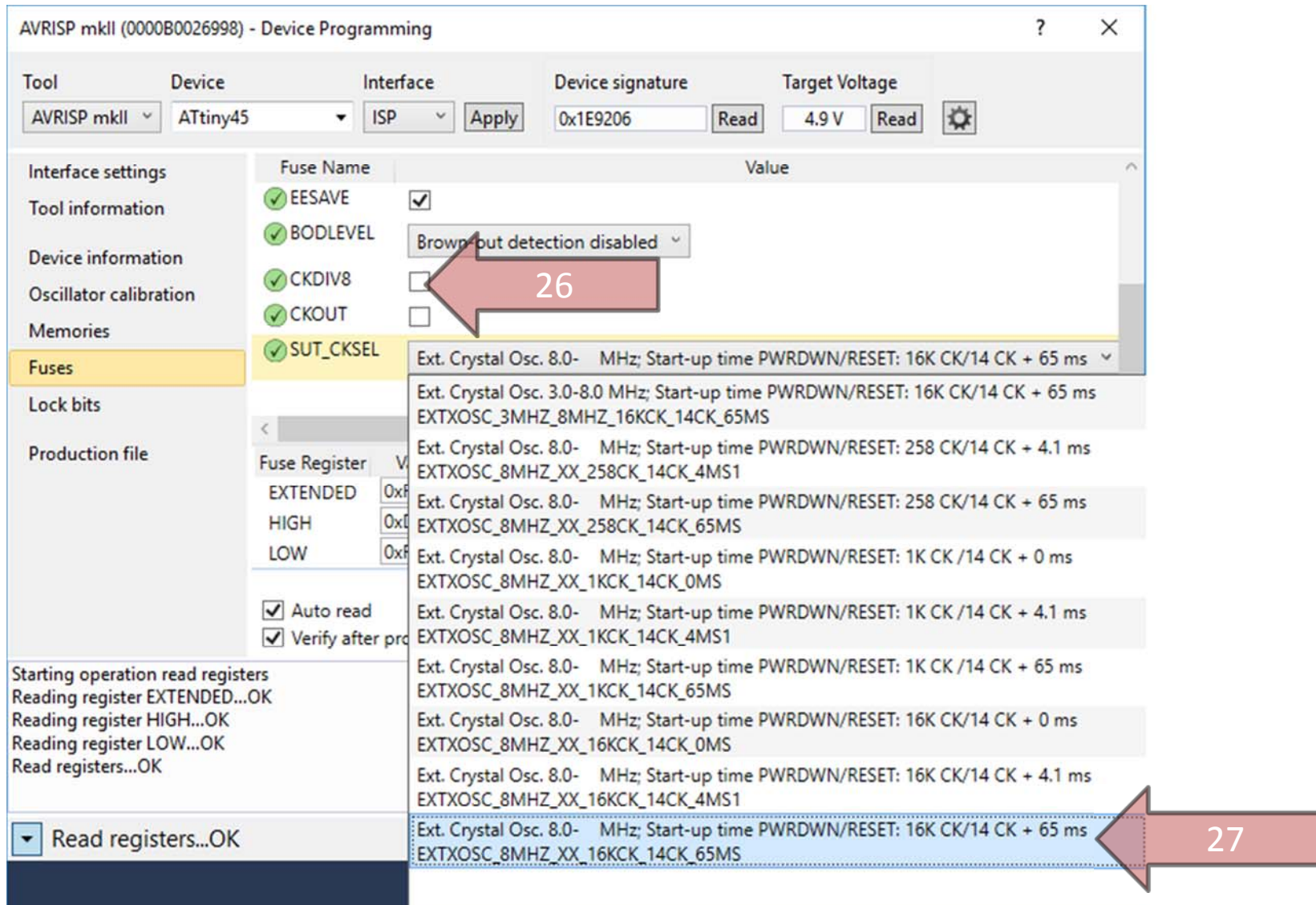
24. Click “Read” under “Device Signature”



If the board was assembled correctly and connected properly, the device signature and voltage values should appear

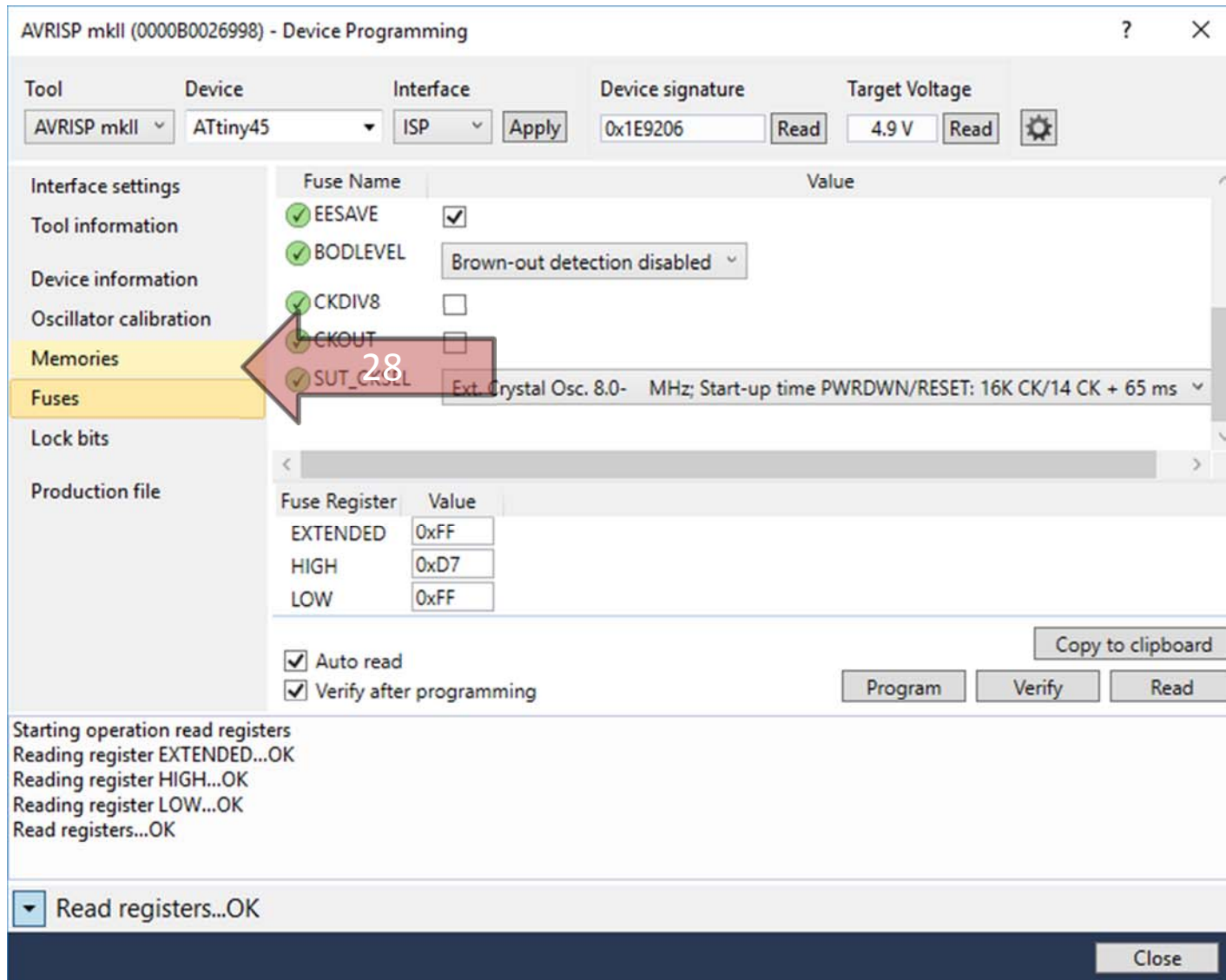


25. Click on “Fuses”

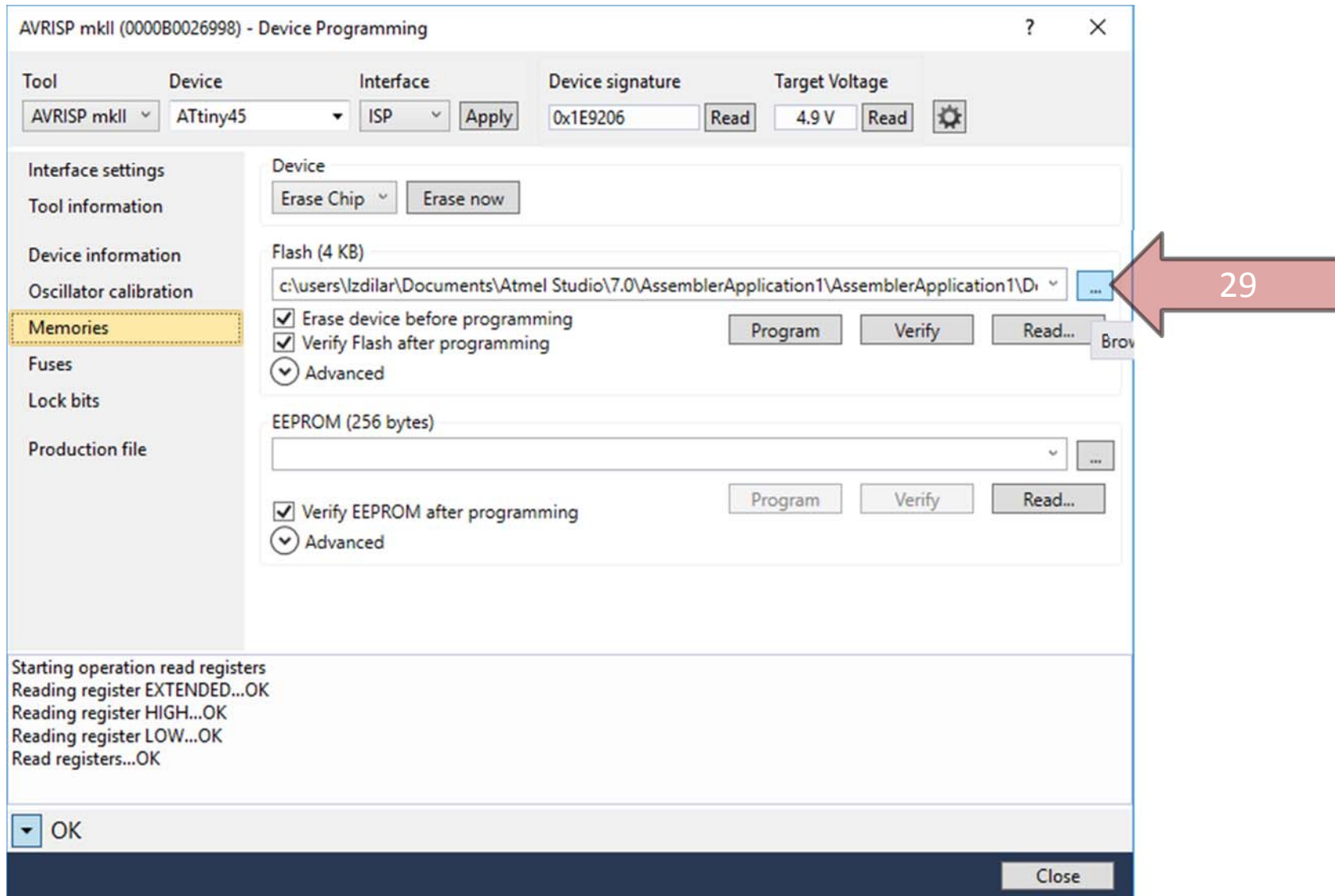


26. Uncheck “CKDIV8”

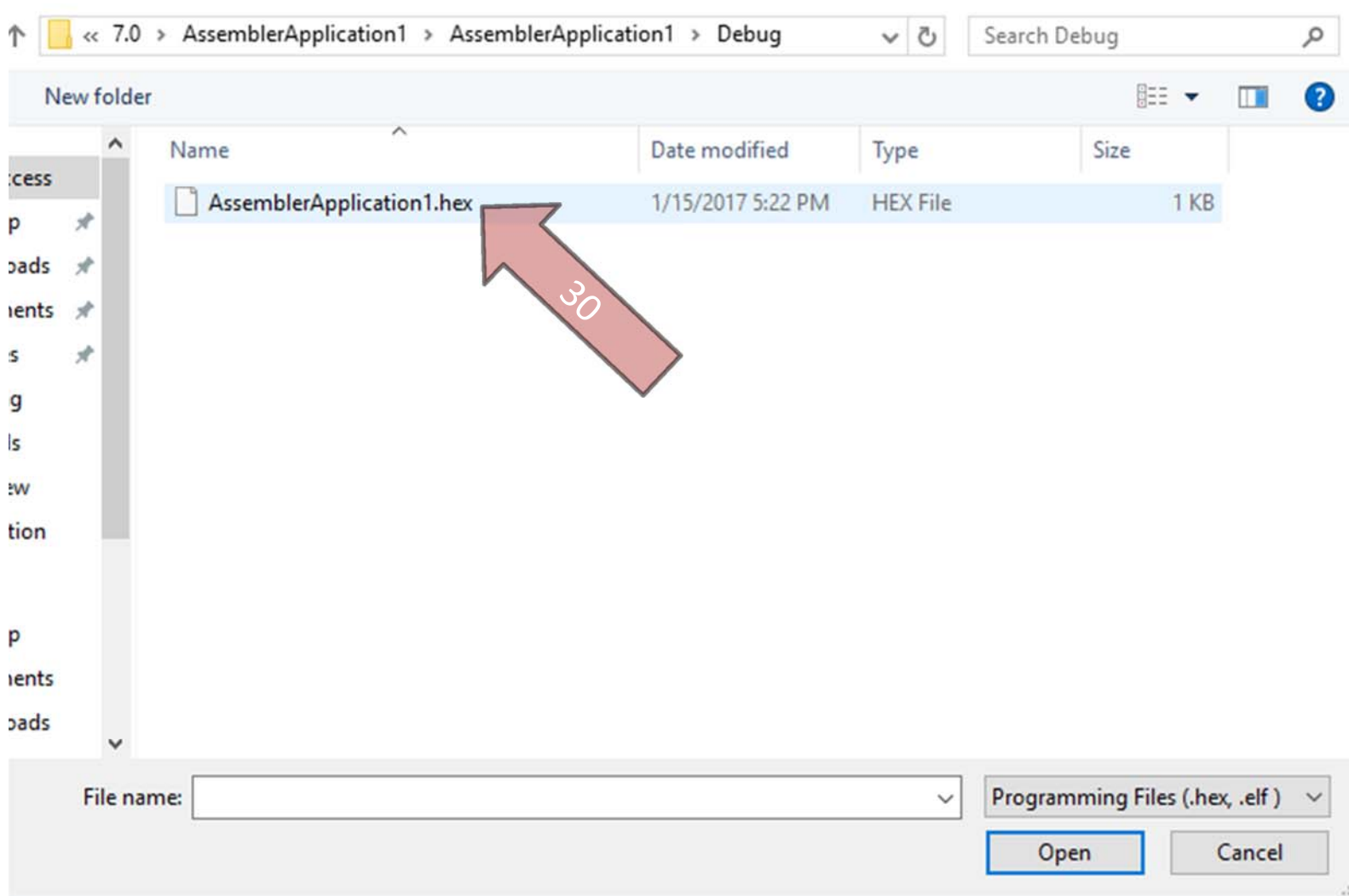
27. Select the highlighted option for “SUT_CKSEL”



28. Click “Memories”

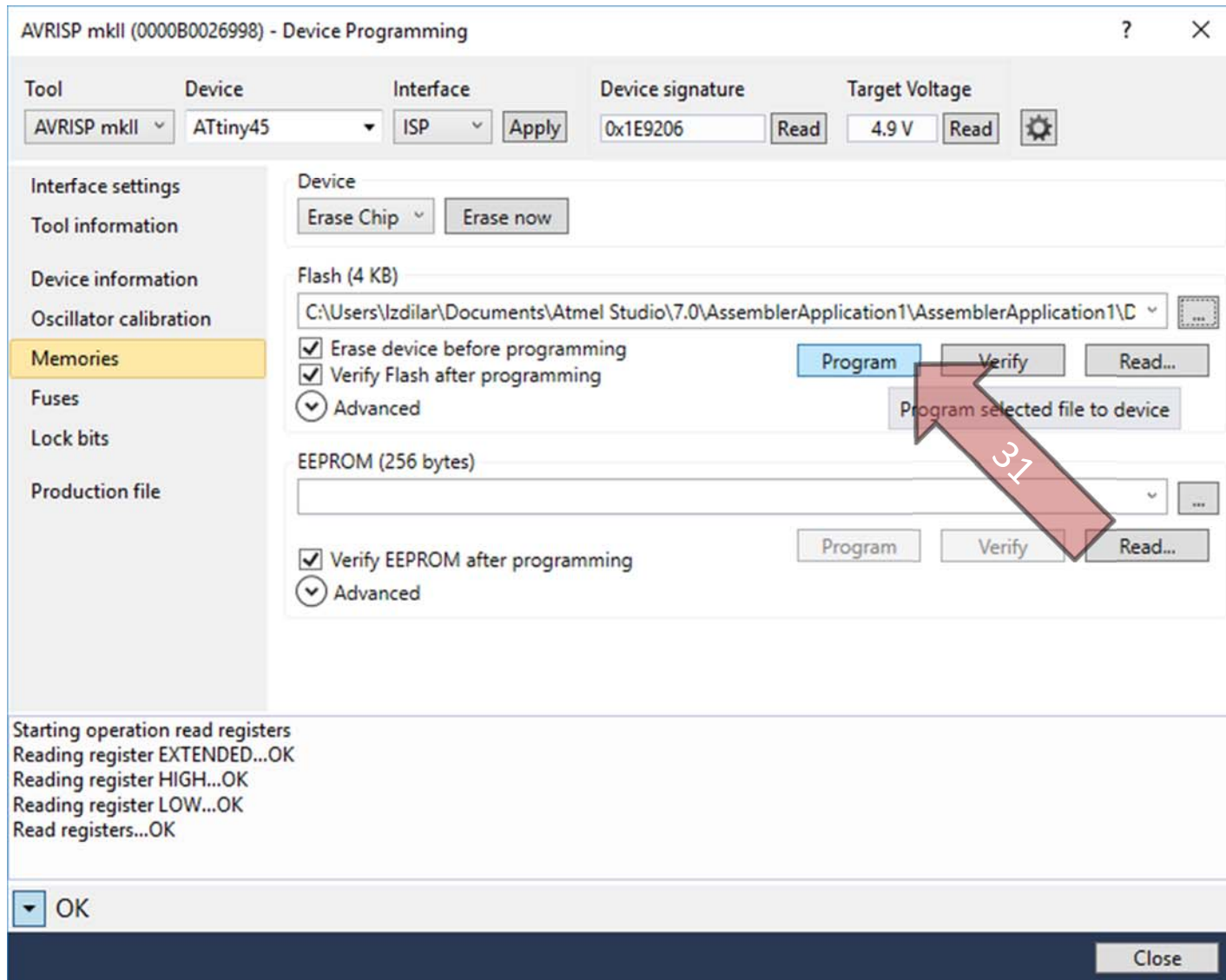


29. Click “...” under “Flash” to select the file to program to the device

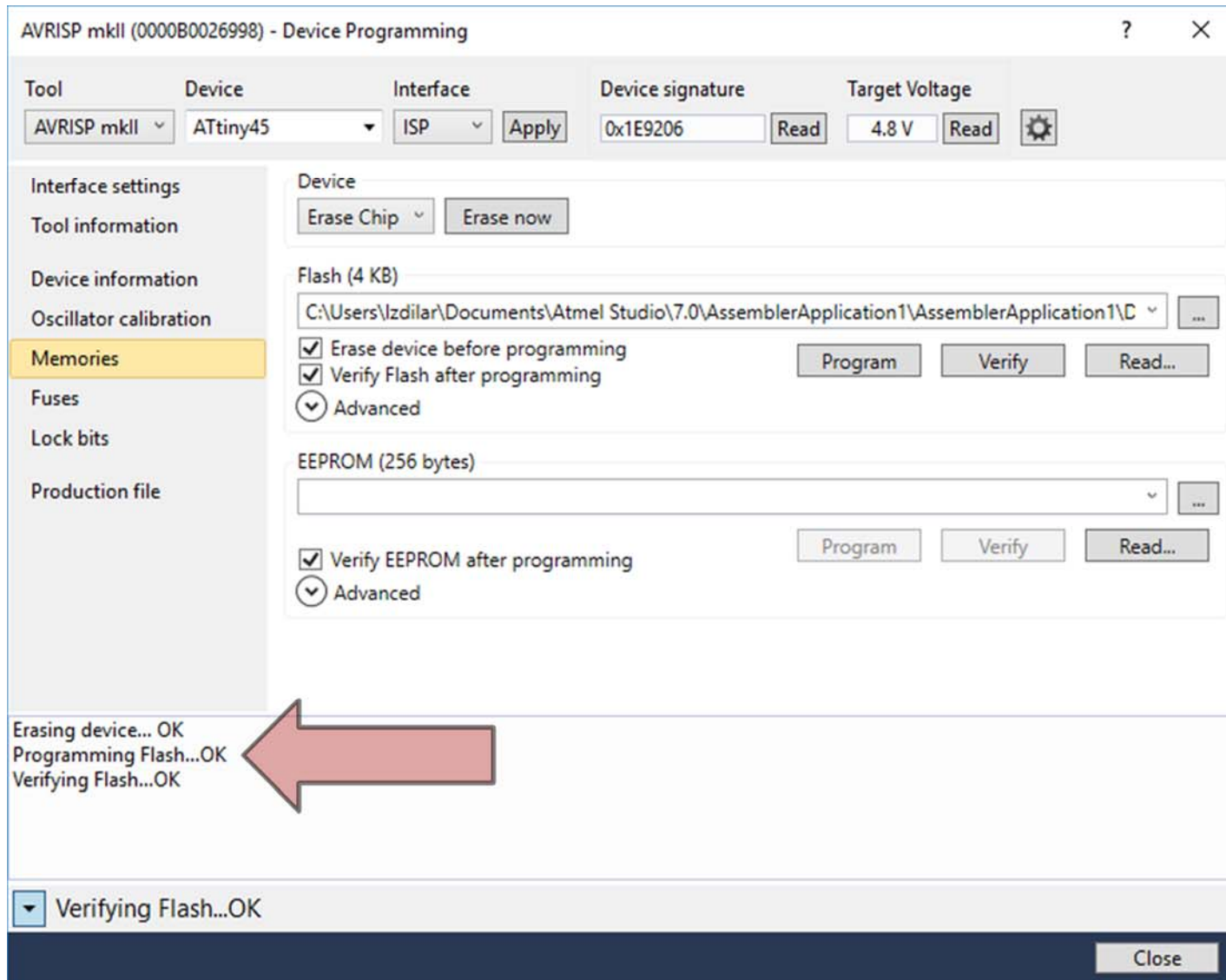


30. Open the compiled .hex file

It can be found in “<ProjectName>\<ProjectName>\Debug”



31. Click “Program”



The messages indicated above should display if the device was programmed successfully