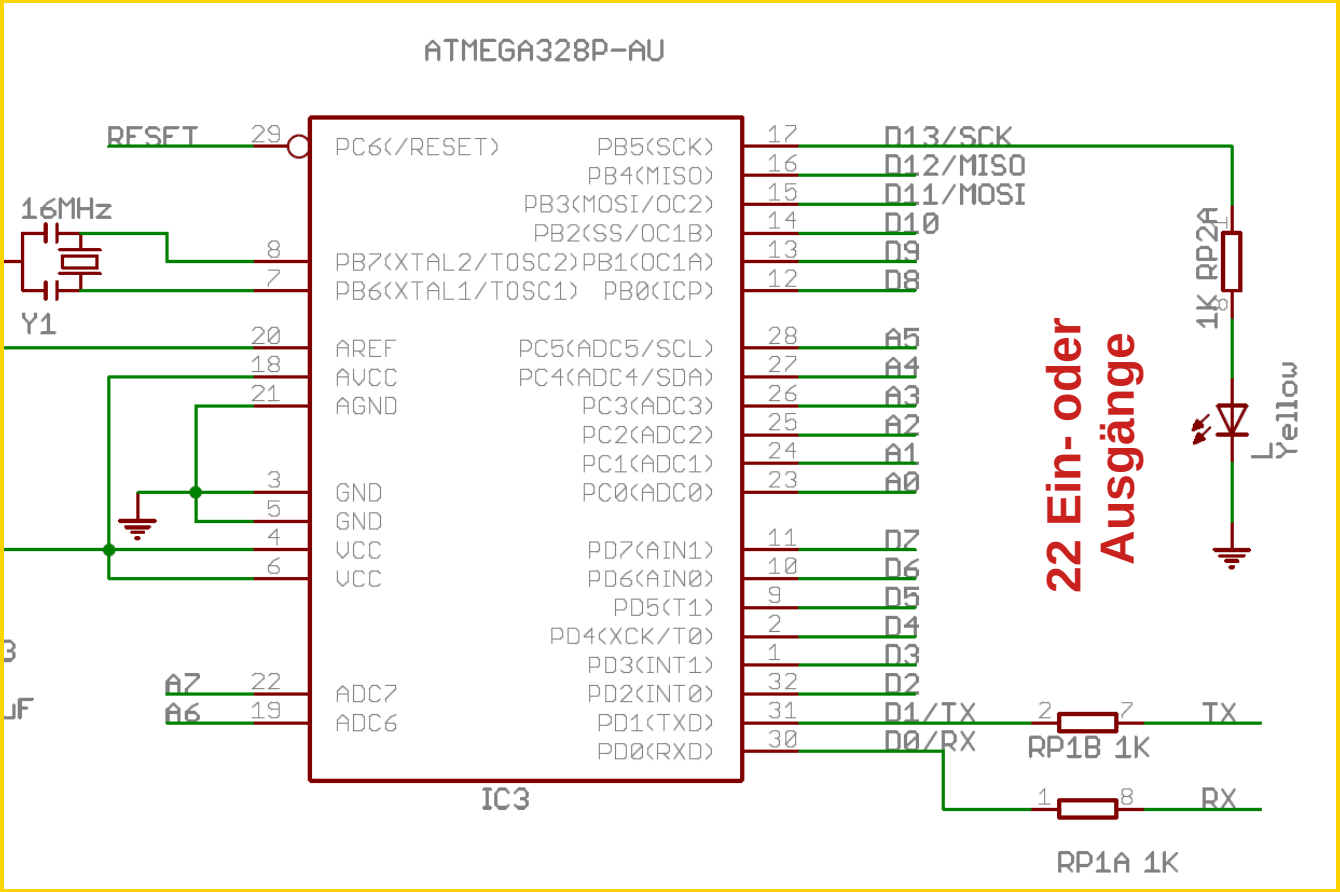
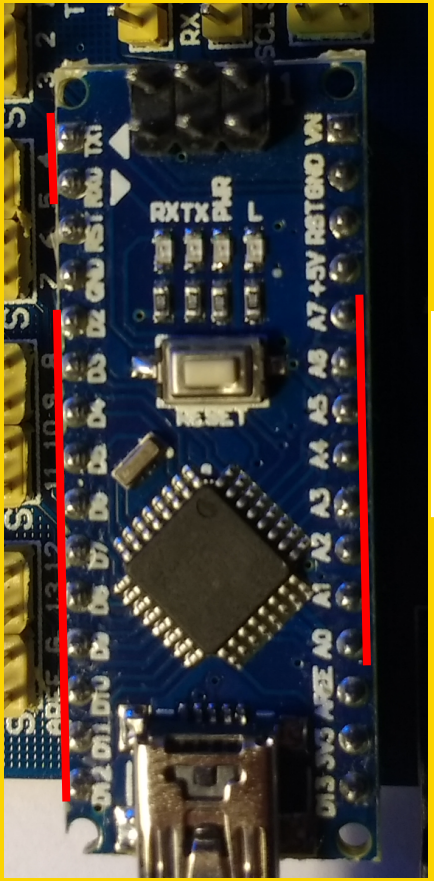
Montage und Programmierung eines Roboters für den Hessen SolarCup Disziplin: SolaRobot Teil 2.0: Hello World

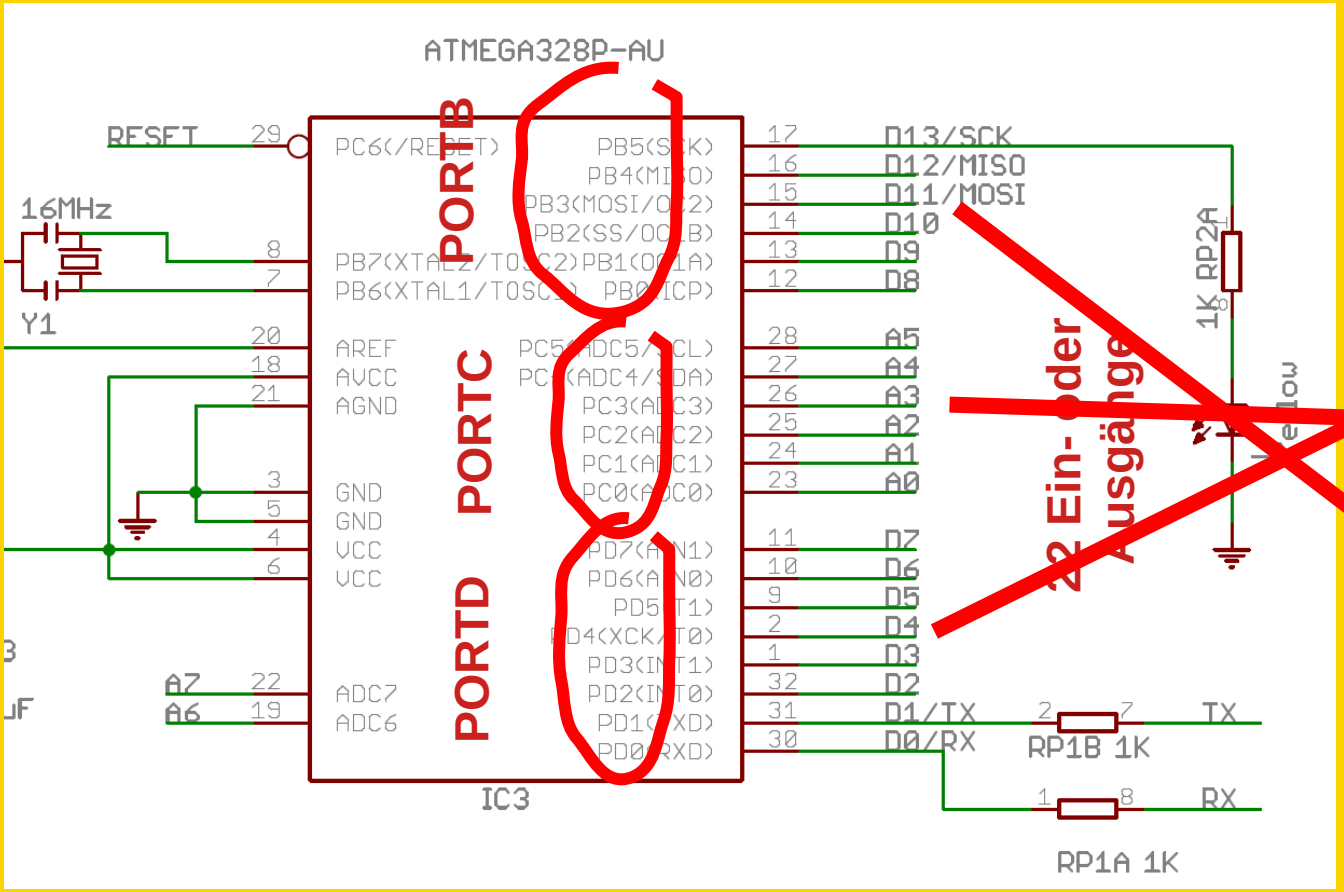
Von Charlotte und Andreas

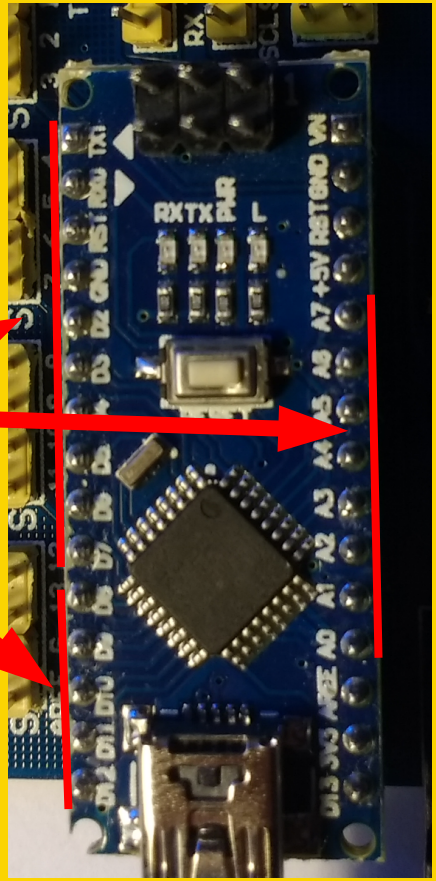
PORTS am Atmega328P





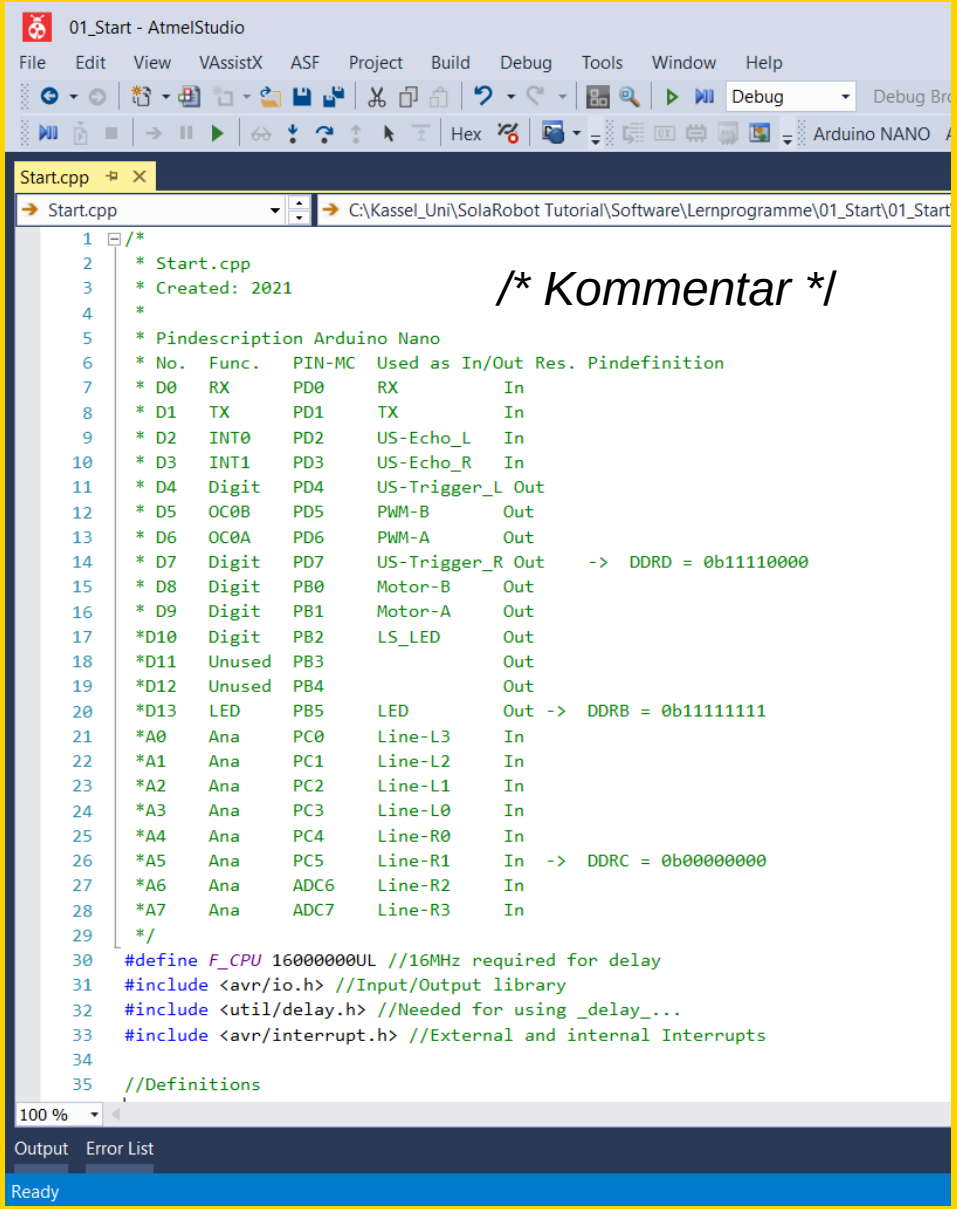
PORTS am Atmega328P





Pinbelegung

D0 und D1 werden für Datenübertragungen gebraucht
D2 Echo vom Ultraschall links
D3 Echo vom Ultraschall rechts
D4 Ultraschall links Triggerleitung
D5 PWM B
D6 PWM ANicht belegt
D7 Ultraschall rechts Triggerleitung
D8 Motorausgang B
D9 Motorausgang A
D10 Liniensensor LEDs on/off
D11 Nicht belegt
D12 Nicht belegt
D13 LED am Nano
A0 Liniensensor links drei
A1 Liniensensor links zwei
A2 Liniensensor links eins
A3 Liniensensor links null
A4 Liniensensor rechts null
A5 Liniensensor rechts eins
A6 Liniensensor rechts zwei
A7 Liniensensor rechts drei



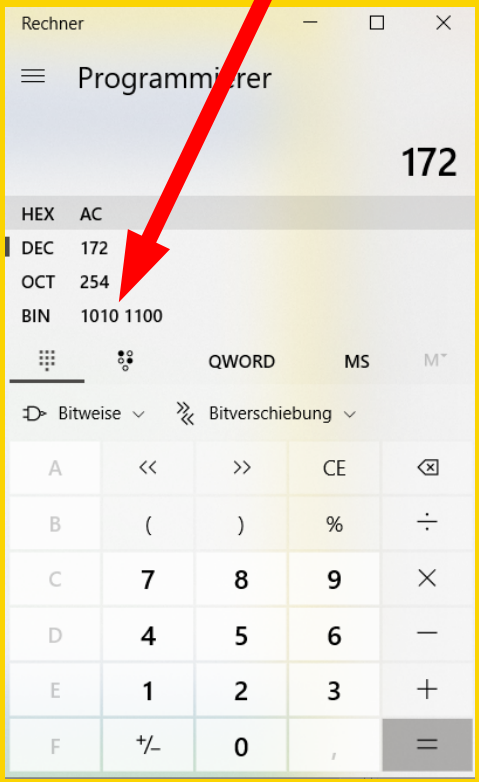

```
/*
 * Start.cpp
 * Created: 2021
 *
 * Pindescription Arduino Nano
 * No.  Func.   PIN-MC  Used as In/Out Res. Pindefinition
 * D0   RX      PD0     RX          In
 * D1   TX      PD1     TX          In
 * D2   INT0    PD2     US-Echo_L   In
 * D3   INT1    PD3     US-Echo_R   In
 * D4   Digit   PD4     US-Trigger_L Out
 * D5   OC0B    PD5     PWM-B       Out
 * D6   OC0A    PD6     PWM-A       Out
 * D7   Digit   PD7     US-Trigger_R Out    ->  DDRD = 0b11110000
 * D8   Digit   PB0     Motor-B     Out
 * D9   Digit   PB1     Motor-A     Out
 *D10   Digit   PB2     LS_LED      Out
 *D11   Unused  PB3                 Out
 *D12   Unused  PB4                 Out
 *D13   LED     PB5     LED         Out ->  DDRB = 0b11111111
 *A0    Ana     PC0     Line-L3     In
 *A1    Ana     PC1     Line-L2     In
 *A2    Ana     PC2     Line-L1     In
 *A3    Ana     PC3     Line-L0     In
 *A4    Ana     PC4     Line-R0     In
 *A5    Ana     PC5     Line-R1     In  ->  DDRC = 0b00000000
 *A6    Ana     ADC6    Line-R2     In
 *A7    Ana     ADC7    Line-R3     In
 */
#define F_CPU 16000000UL //16MHz required for delay
#include <avr/io.h> //Input/Output library
#include <util/delay.h> //Needed for using _delay_...
#include <avr/interrupt.h> //External and internal Interrupts

//Definitions
```

unsigned char

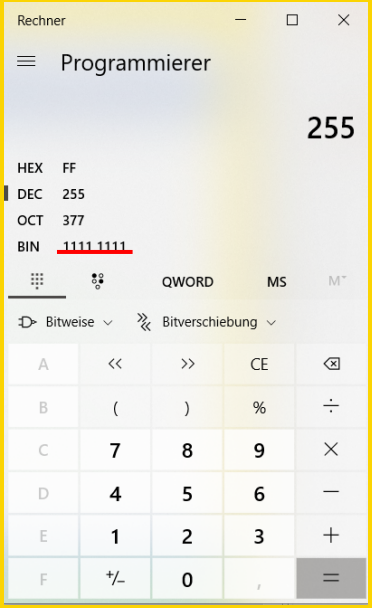
1 Byte bestehend aus 8 Bit
z.B.:

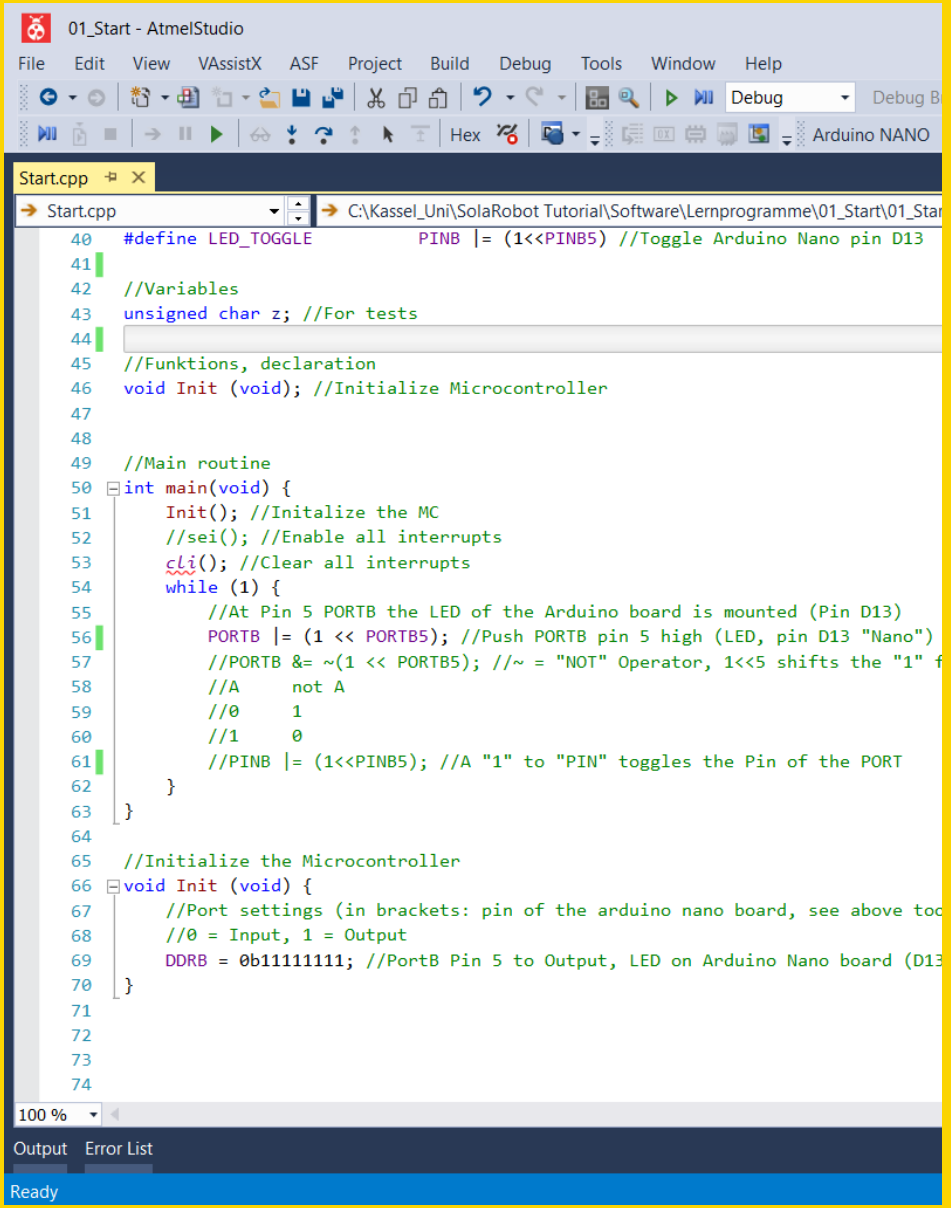
1	0	1	0	1	1	0	0

= 172



0 bis 255





void Funktionsname (void)

z.B.: void Init (void)

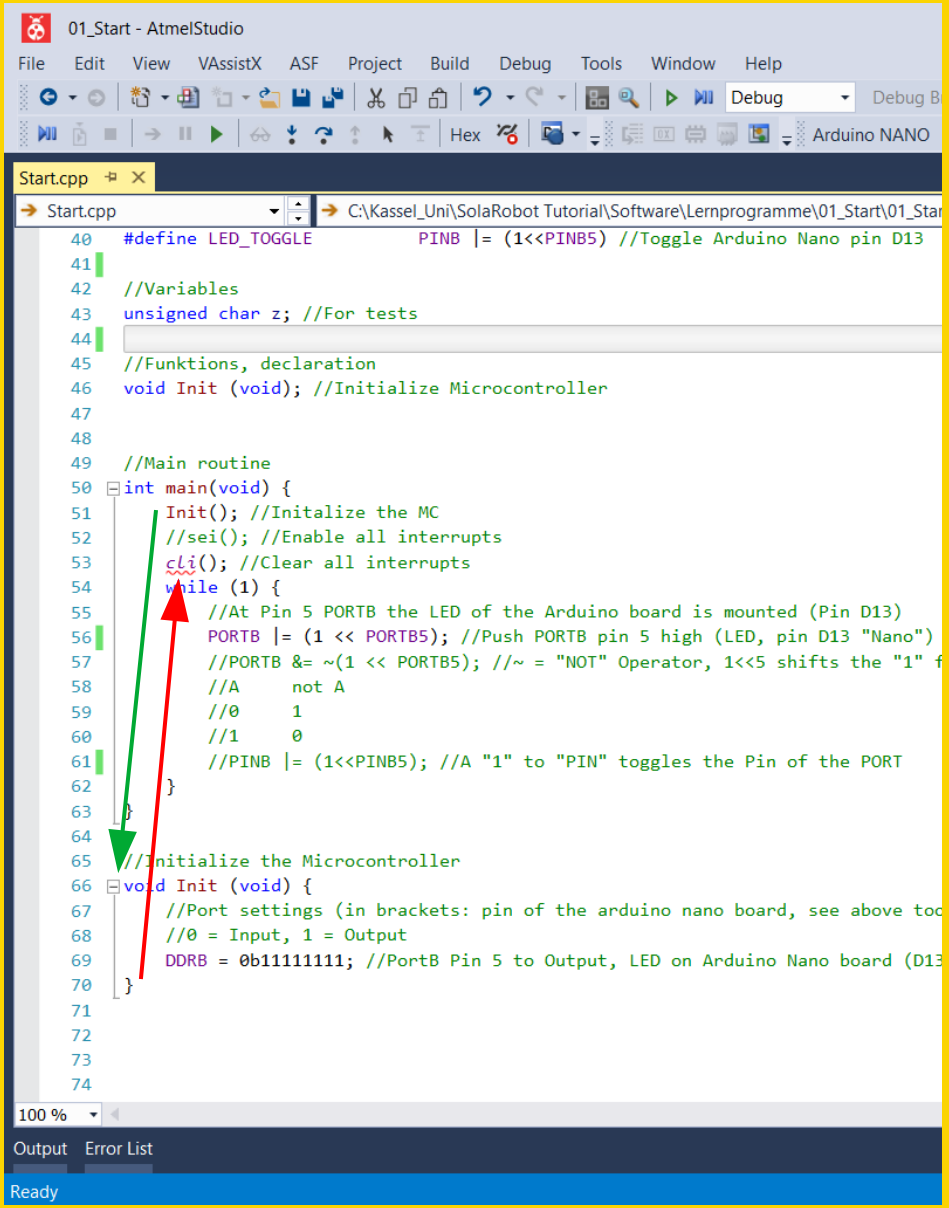
Deklaration für den Compiler im
oberen Teil, vor der Hauptfunktion(main).
Bekanntmachung der Funktion
für den Compiler.

Aufruf im Hauptprogramm (oder woanders)

Definition im unteren Teil

Wenn die Funktion aufgerufen wird
Init();
springt der Mikrocontroller dort hin,
arbeitet die Funktion ab und springt an
die Ausgangsstelle zurück.

void → nichts



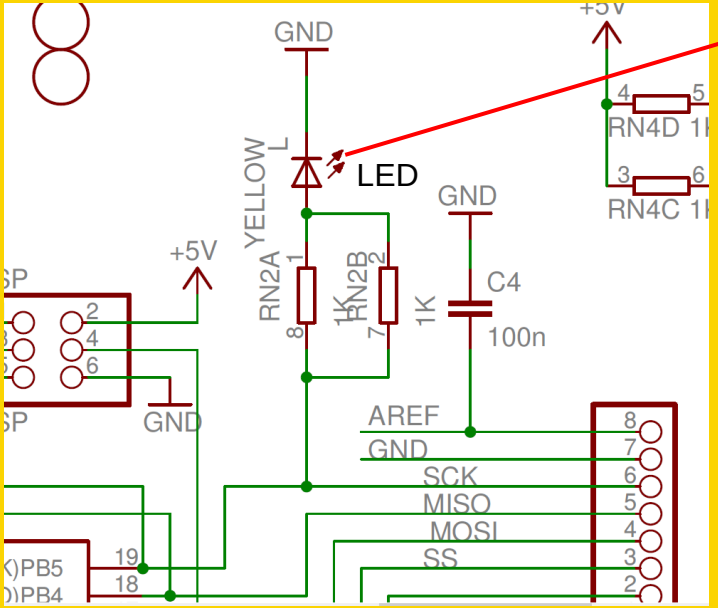
DDRB:

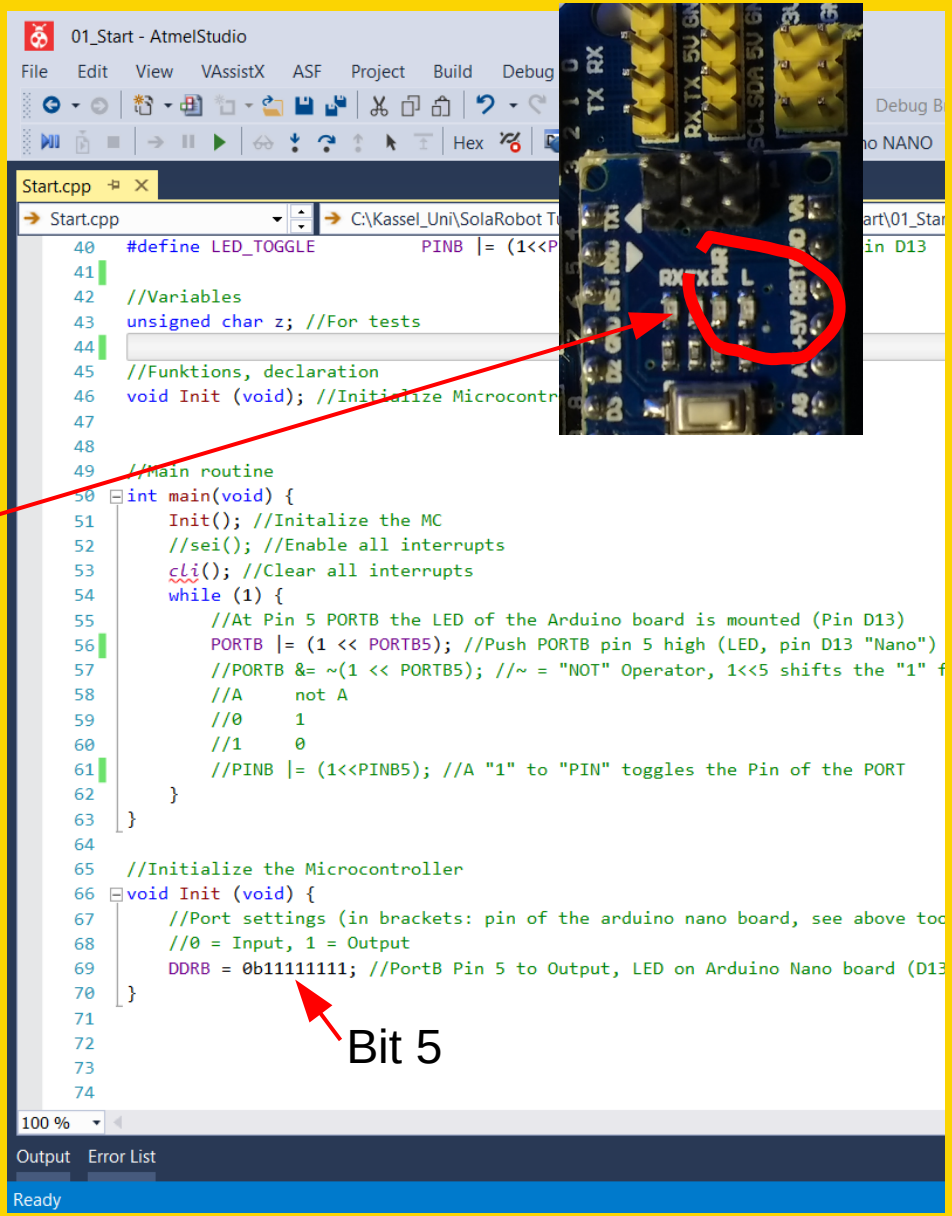
Data Direction Register für PORTB

0b… Die folgende Ziffernfolge gibt eine Binärzahl wieder.

0b 1 1 1 1 1 1 1 1

Bit 5 Bit 0



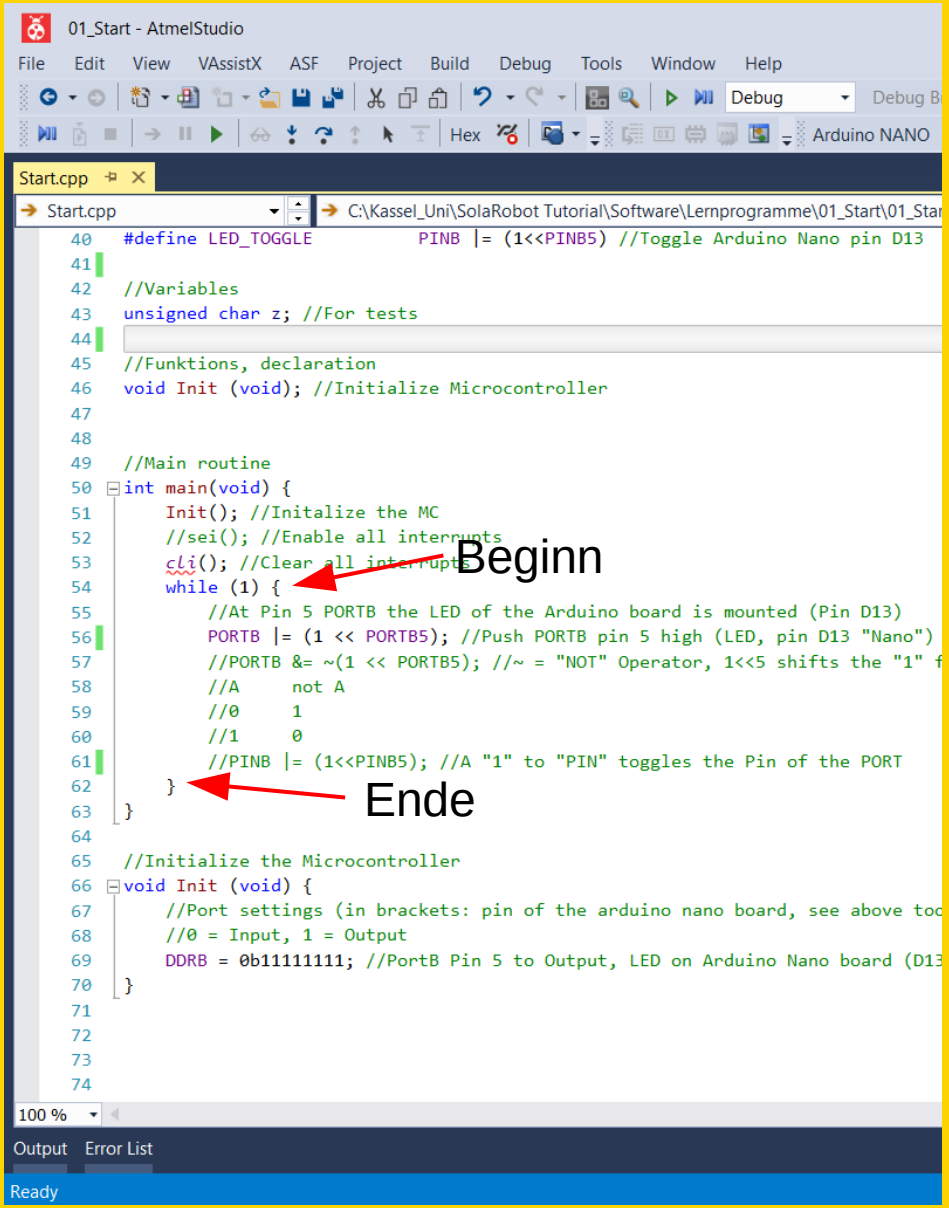


Int main (void):

Der MC beginnt immer mit der main-Funktion

while:

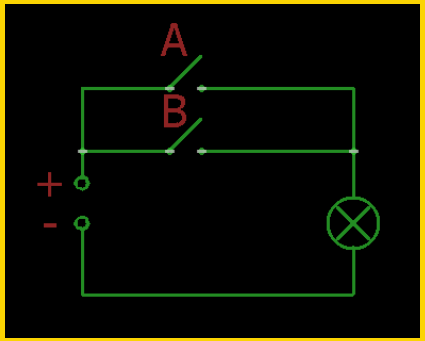
Solange die Bedingung in den nachfolgenden runden Klammern erfüllt ist, wiederhole die Anweisungen zwischen den geschweiften Klammern der „while-Schleife“.
Eine Bedingung gilt dann als erfüllt, wenn sie
NICHT NULL
ist.



Wahrheitstabelle:

Oder (Or |)

A	B	A | B
0	0	0
0	1	1
1	0	1
1	1	1



0	0	0	0	0	0	0	0	= 0

oder

0	0	0	1	0	0	0	0	= 16

ergibt

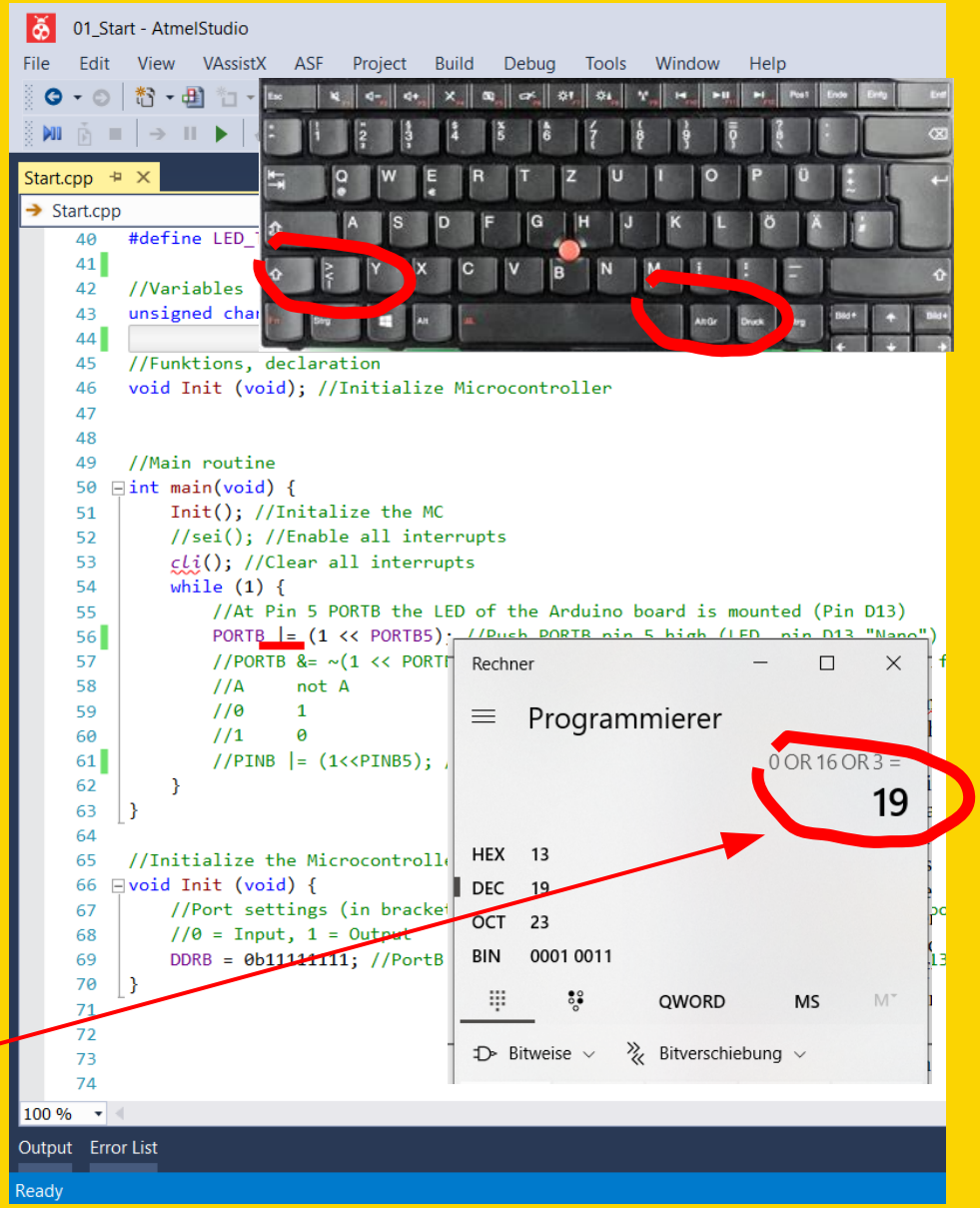
0	0	0	1	0	0	0	0	= 16

oder

0	0	0	0	0	0	1	1	= 3

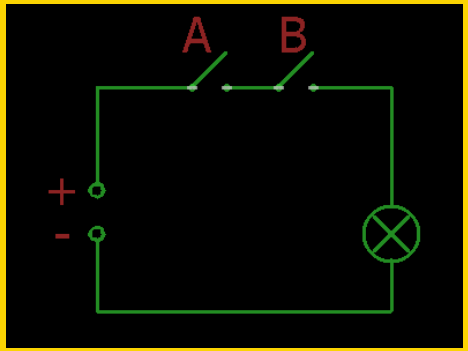
ergibt

0	0	0	1	0	0	1	1	= 19



Wahrheitstabelle:
Und (And &)

A	B	A & B
0	0	0
0	1	0
1	0	0
1	1	1



0	0	0	1	0	0	1	1

= 19

und

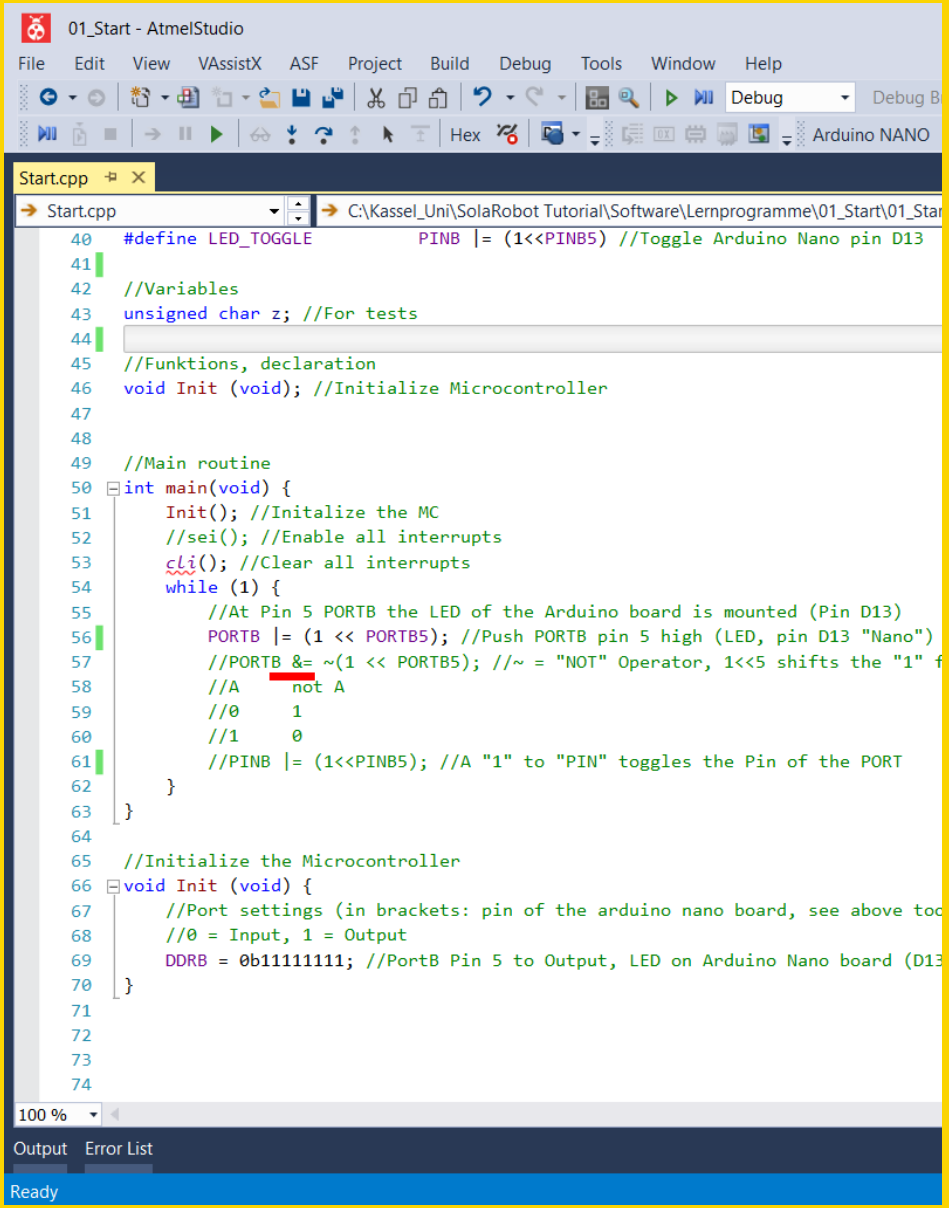
1	1	1	0	1	1	1	0

= 238

gleich

0	0	0	0	0	0	1	0

= 2



<< Operator:

Verschiebt ein Bit nach links

0	0	0	0	0	0	0	0

= 0

(1 << 1) eine 1 einmal nach links schieben

0	0	0	0	0	0	1	0

= 2

(1 << 4) eine 1 viermal nach links schieben

0	0	0	1	0	0	0	0

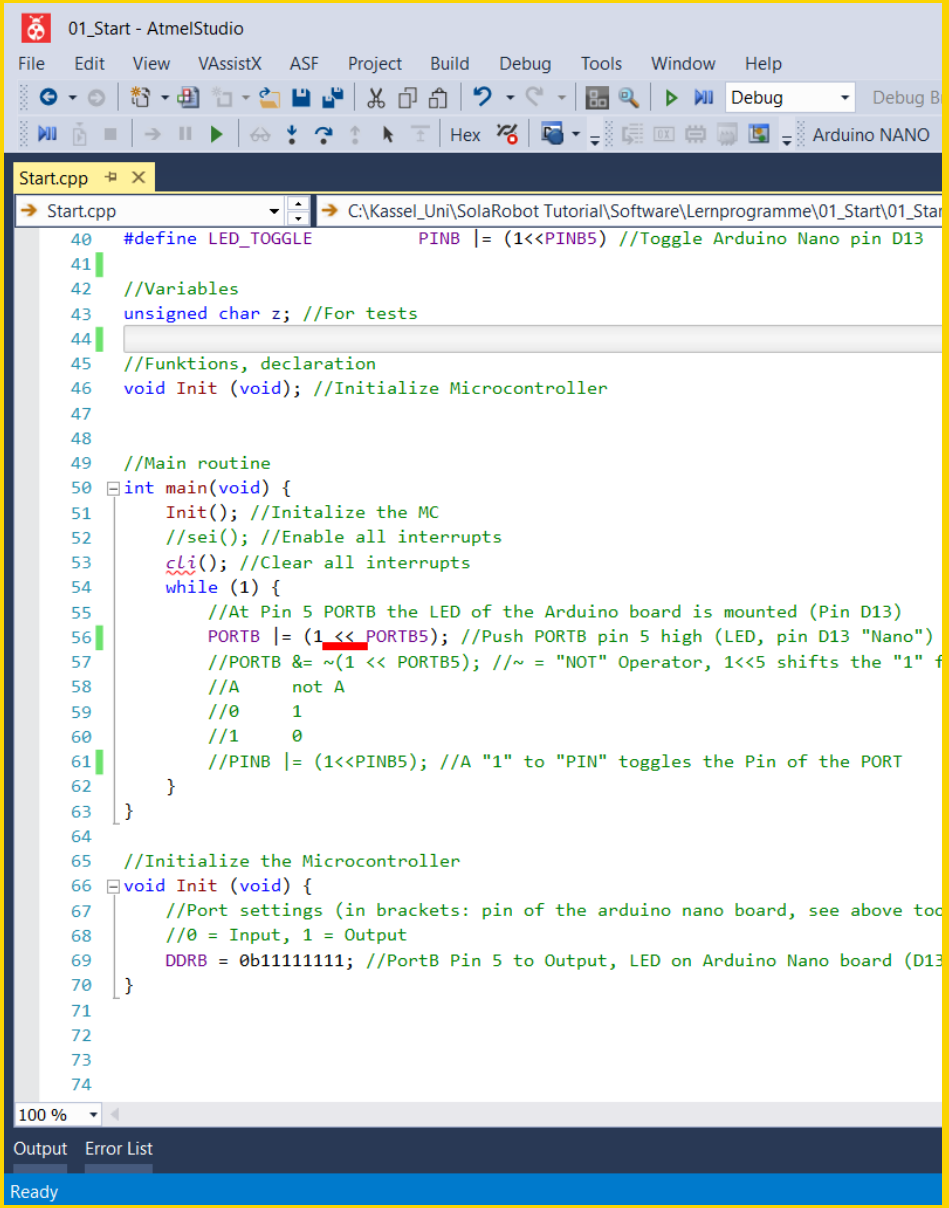
= 16

| (1 << 6)

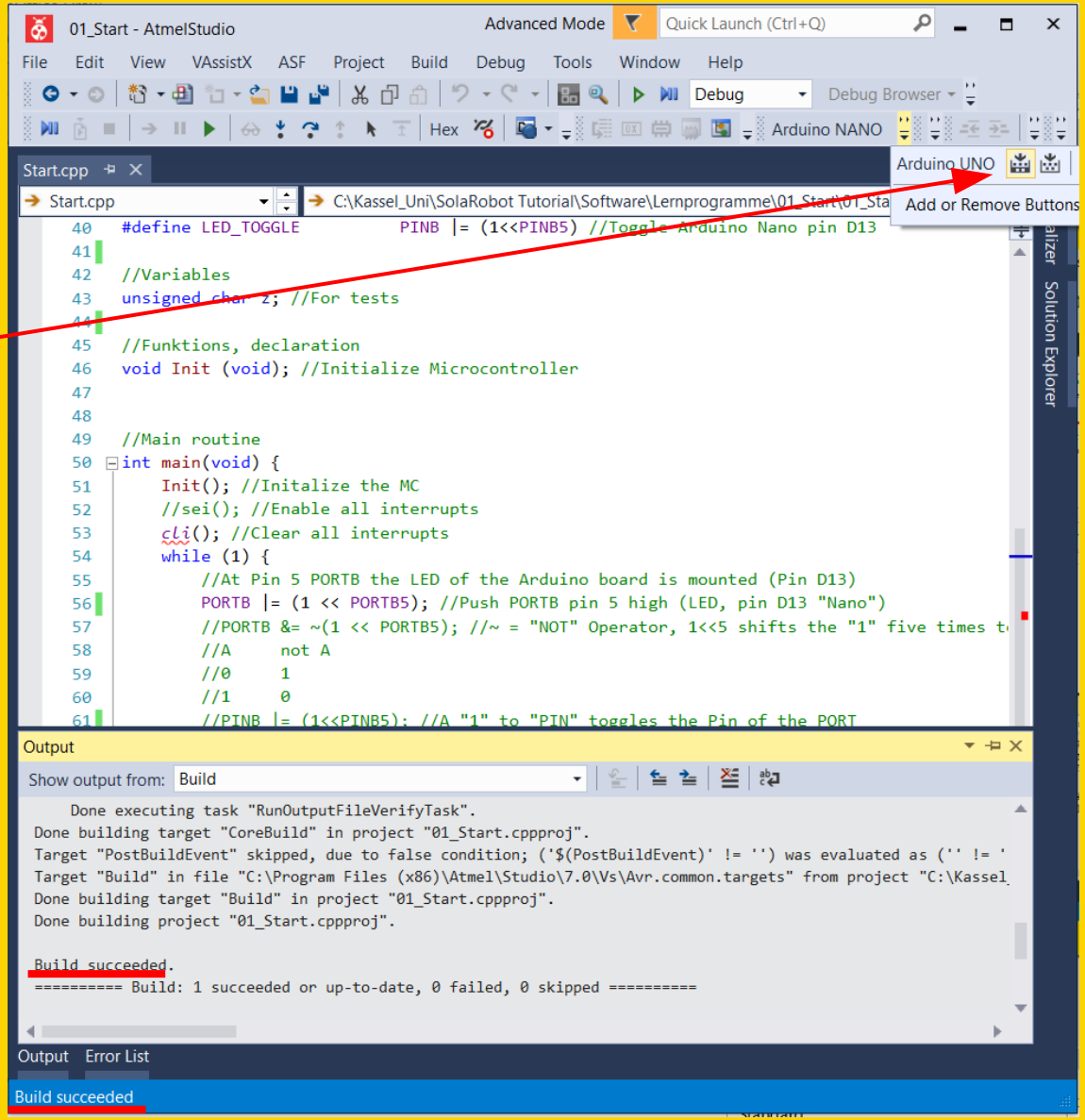
0	1	0	1	0	0	0	0

= 80

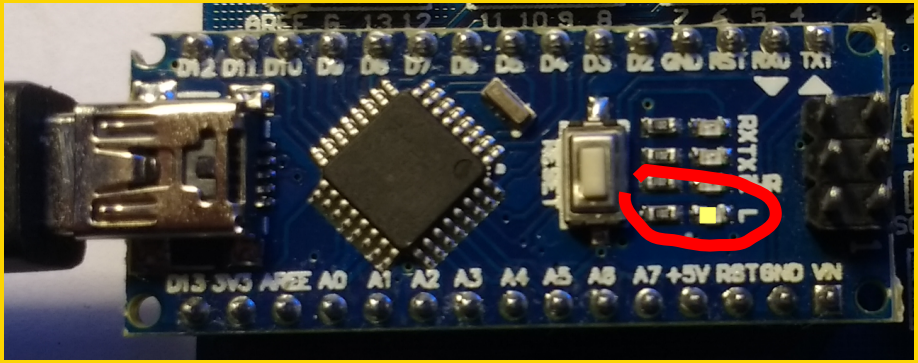
PORTB |= (1 << PORTB5) eine Abkürzung für
PORTB = PORTB | (1<<PORTB5)

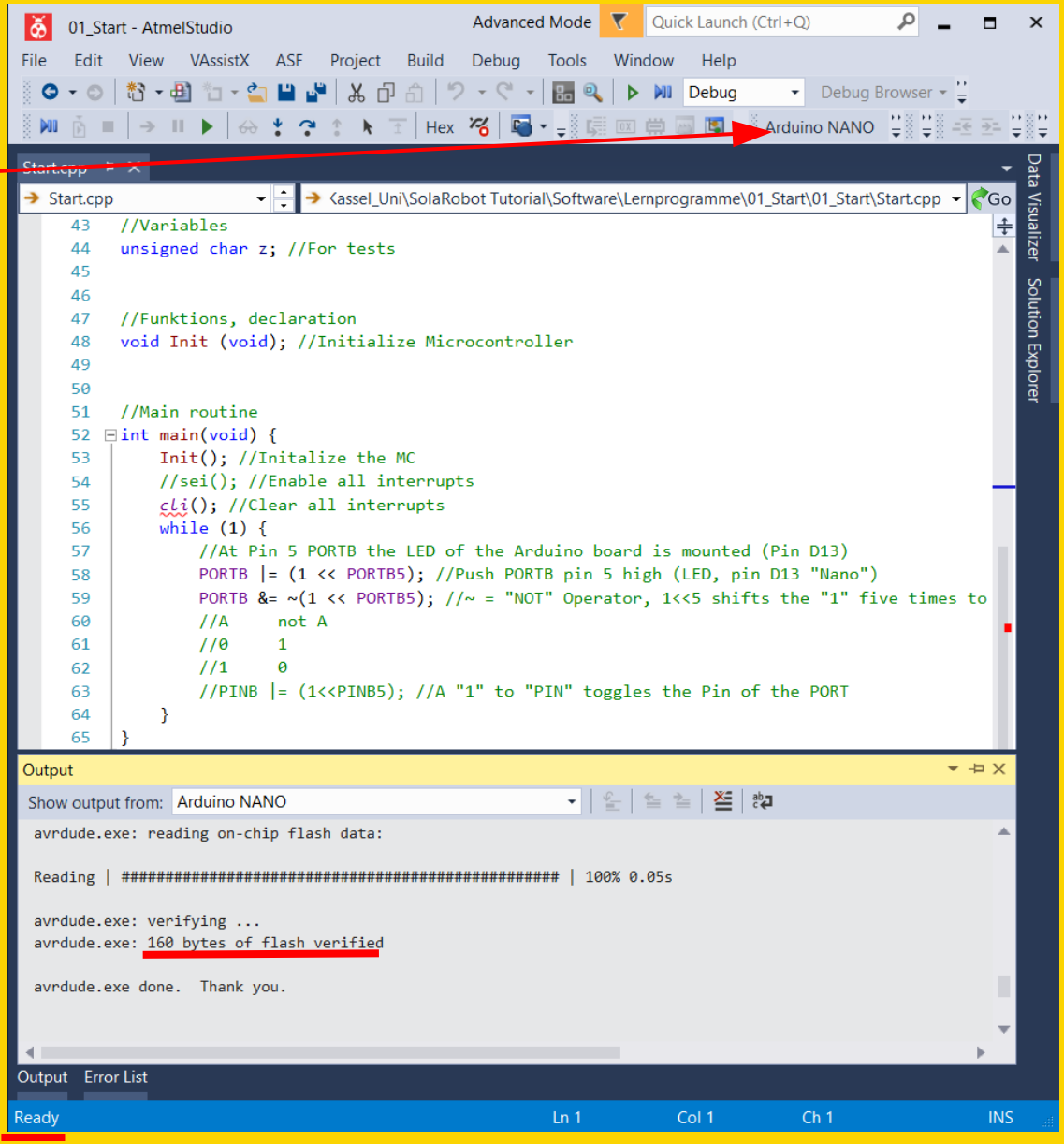


Compilieren:
das Programm, das wir verstehen,
wird umgewandelt in ein Programm,
das der Mikrocontroller versteht.

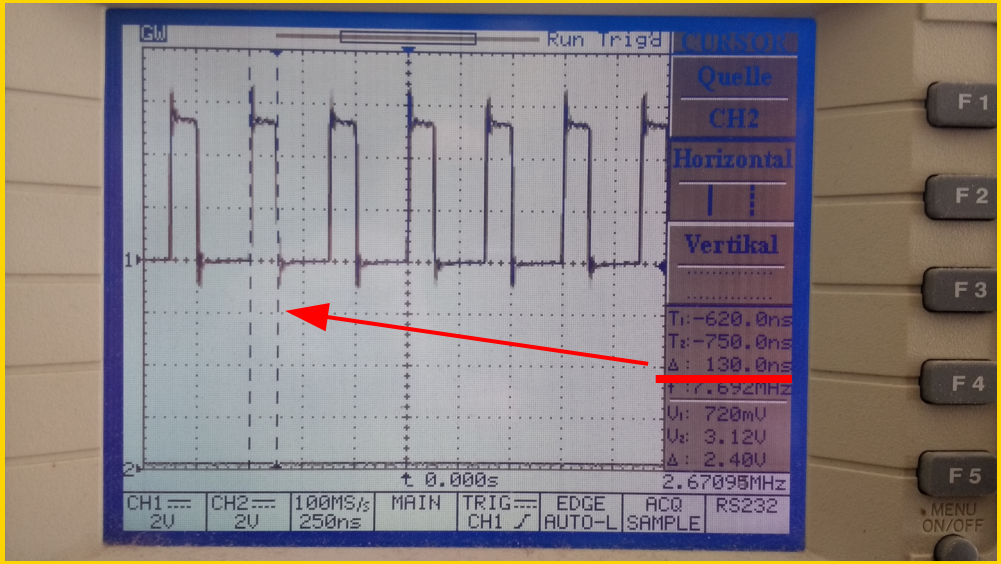


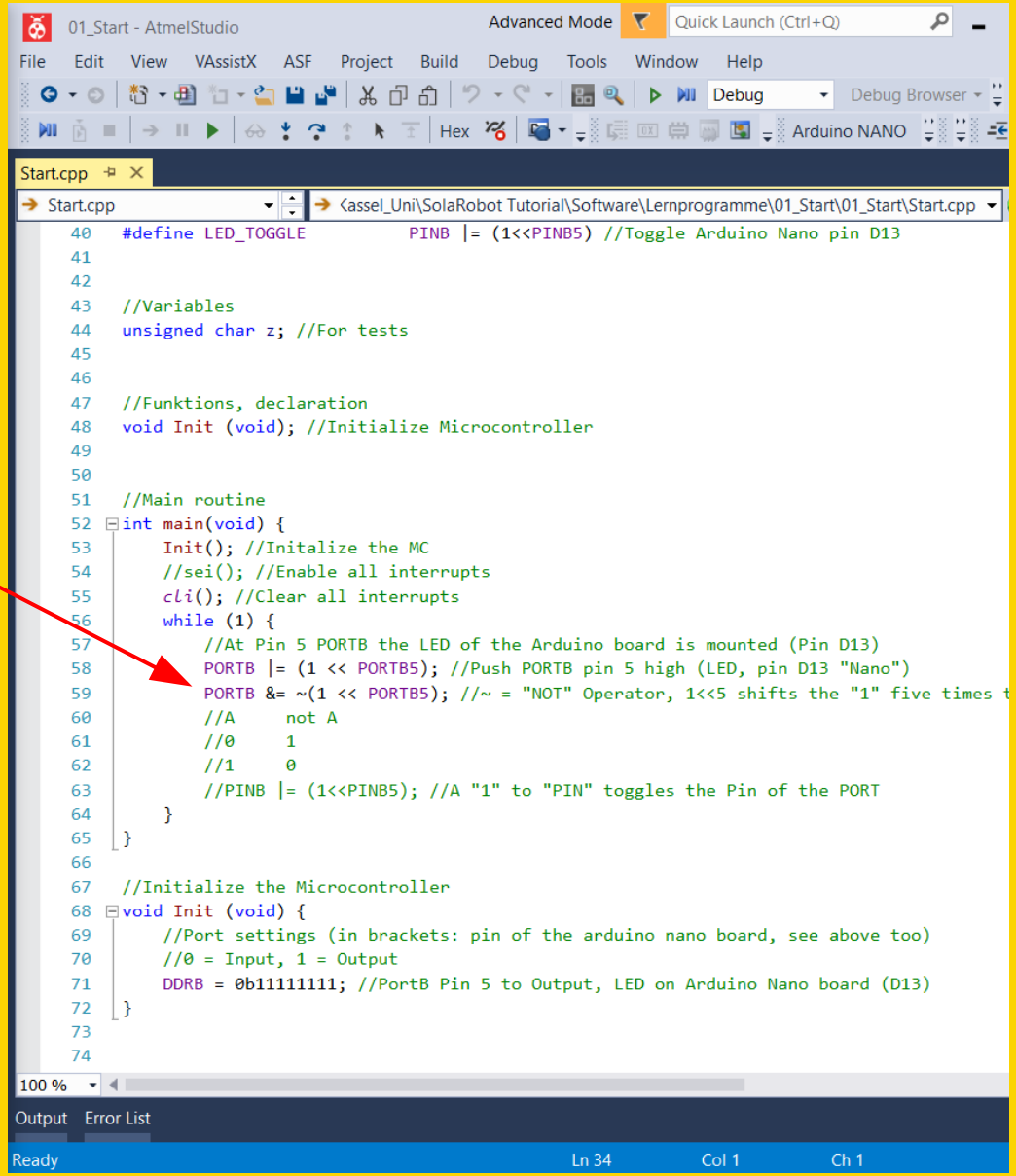
Flashen:
das Programm auf den Mikrocontroller downloaden.





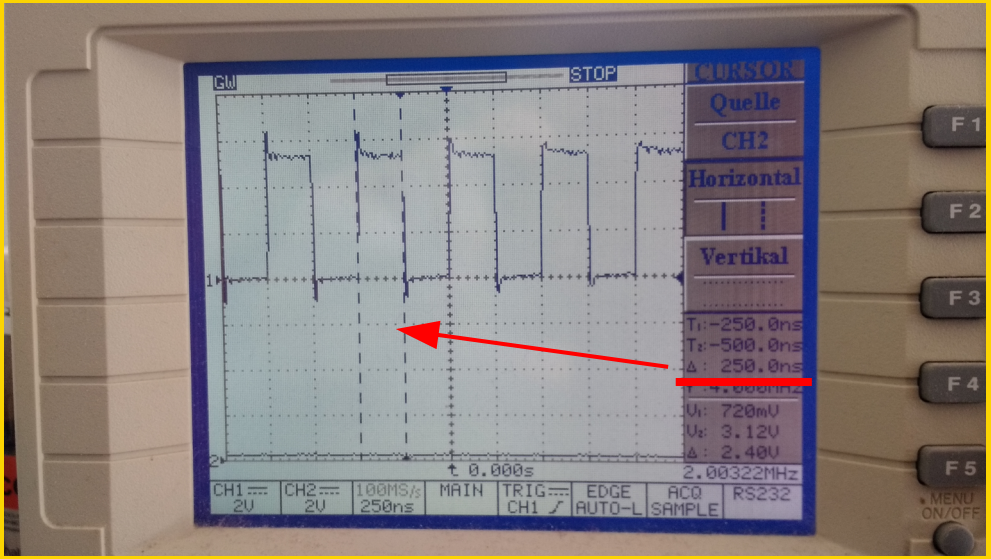
Kommentar aufgehoben:
Eine Zeile ins Programm aufnehmen.





Auskommentieren:
Eine Zeile aus dem Programm nehmen.

Kommentar aufheben:
Eine Zeile ins Programm aufnehmen.

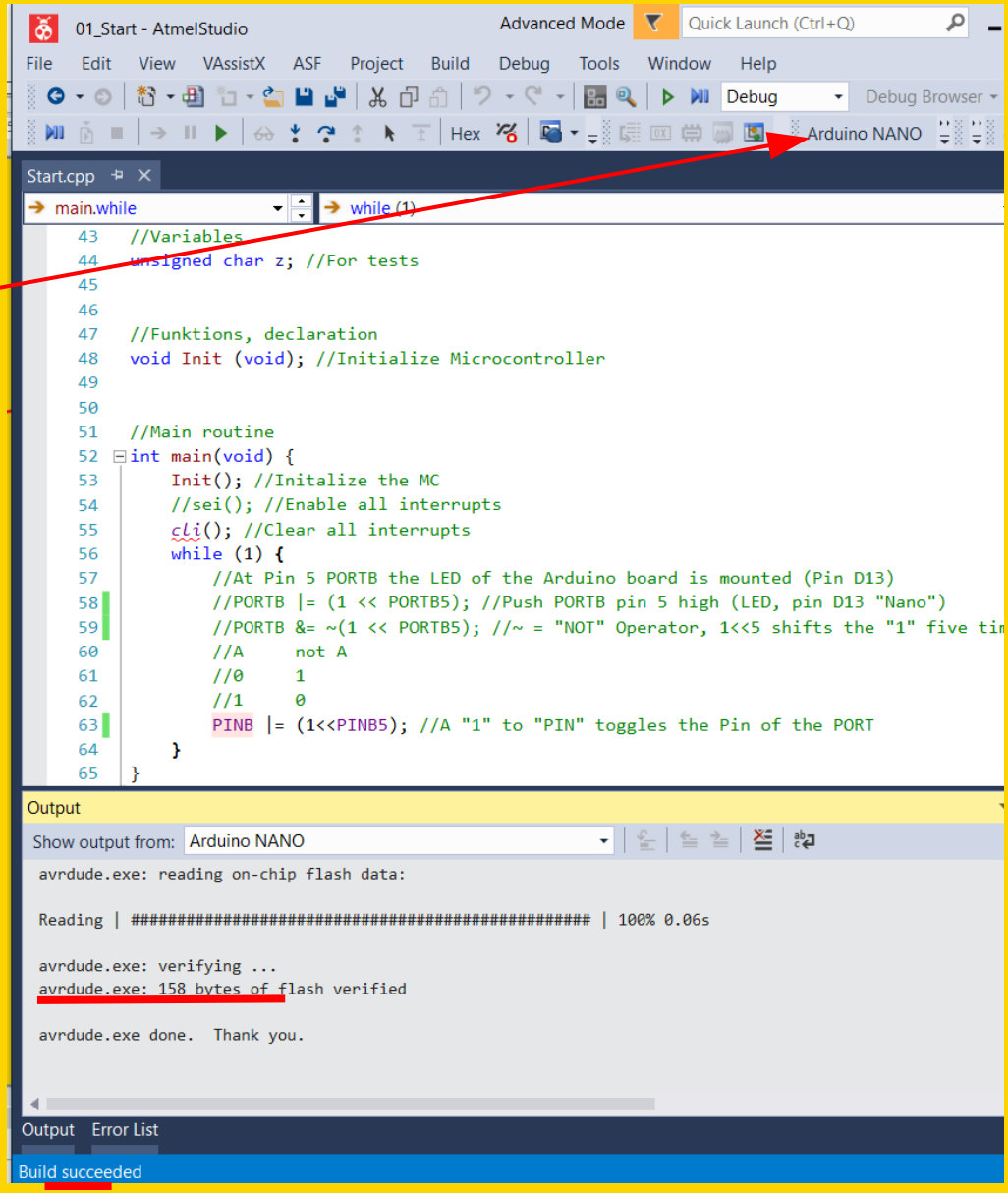



```
01_Start - AtmelStudio
Advanced Mode
Quick Launch (Ctrl+Q)
File Edit View VAssistX ASF Project Build Debug Tools Window Help
Debug
Hex
Arduino NANO
Start.cpp*
main.while
while (1)
```

```
43  //Variables
44  unsigned char z; //For tests
45
46
47  //Funktions, declaration
48  void Init (void); //Initialize Microcontroller
49
50
51  //Main routine
52  int main(void) {
53      Init(); //Initalize the MC
54      //sei(); //Enable all interrupts
55      cli(); //Clear all interrupts
56      while (1) {
57          //At Pin 5 PORTB the LED of the Arduino board is mounted (Pin D13)
58          //PORTB |= (1 << PORTB5); //Push PORTB pin 5 high (LED, pin D13 "Nano")
59          //PORTB &= ~(1 << PORTB5); //~ = "NOT" Operator, 1<<5 shifts the "1" fiv
60          //A     not A
61          //0     1
62          //1     0
63          PINB |= (1<<PINB5); //A "1" to "PIN" toggles the Pin of the PORT
64      }
65  }
66
67  //Initialize the Microcontroller
68  void Init (void) {
69      //Port settings (in brackets: pin of the arduino nano board, see above too)
70      //0 = Input, 1 = Output
71      DDRB = 0b11111111; //PortB Pin 5 to Output, LED on Arduino Nano board (D13)
72  }
73
74
75
76
```

```
100 %
Output  Error List
Ready   Ln 63   Col 9   Ch 3
```

Flashen:
das Programm auf den Mikrocontroller downloaden.



Montage und Programmierung eines Roboters für den Hessen SolarCup Disziplin: SolaRobot Teil 2.1: Hello World_2

Von Charlotte und Andreas