

*Montage und Programmierung
eines Roboters für
ROBOCUP JUNIOR RESCUE
mit Arduino Nano
Teil 2.3: Data Visualizer*

Solution Explorer

Transmit_literal

C++

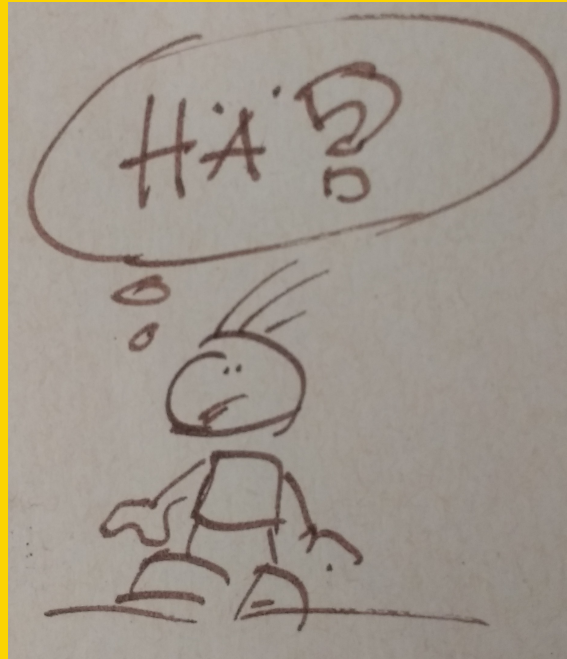
Header

USART

for

Integer

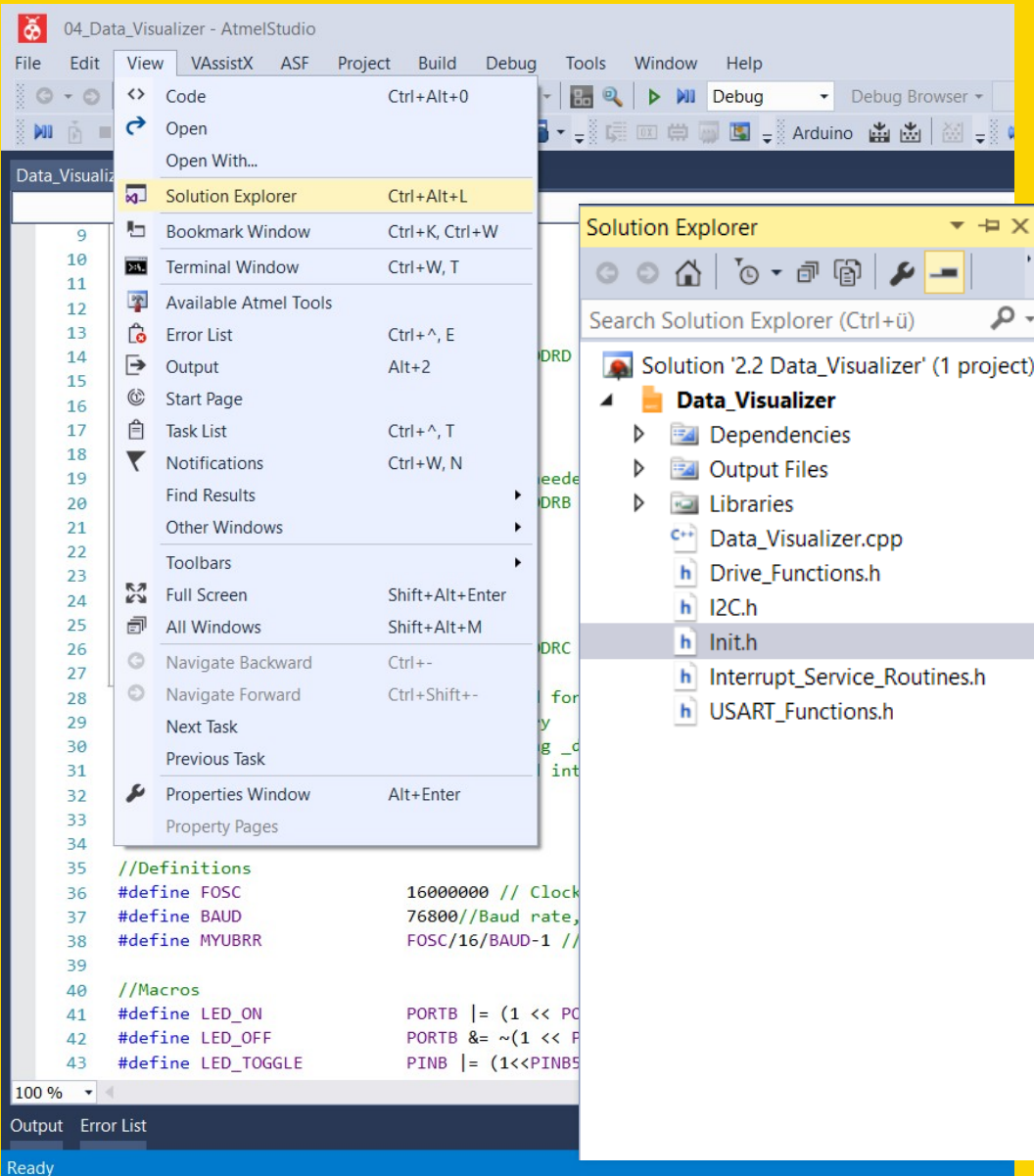
Init.h



Array

Data Visualizer

Data[7]



Header:

Init.h

USART_Functions.h

```

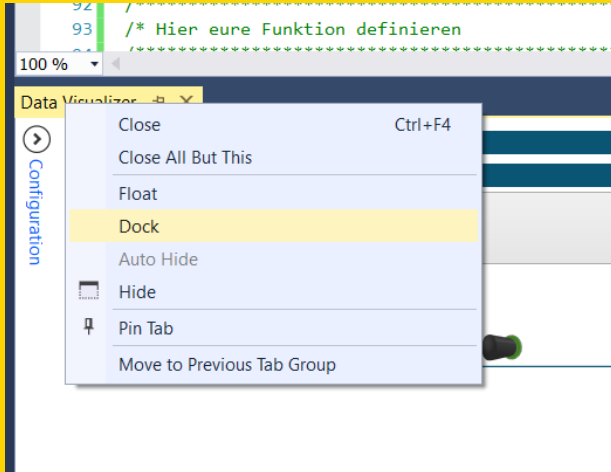
Data_Visualizer.cpp  Init.h  USART_Functions.h
INIT_H
#define INIT_H

16 //Initialize the Microcontroller
17
18 void Init (void) {
19     //Port settings
20     //0 = Input, 1 = Output
21     DDRB = 0b00111111; //00n.n,1LED,1Servo,1Motor_B,1PWM_B,1PWM_A,1MotorA
22     DDRC = 0b00000000; //00n.n,0SCL,0SDA,1Push_1,111Dist-4 to -2
23     DDRD = 0b10000000; //1US_Trig,0Push_2,0BLnWH,0Colour,0Encoder,0US_Echo,
24     PORTD = 0b00001000; //Pull Up Resistor for encoder
25     PORTC = 0b00001000; //Pull Up Resistor for Push_1
26     //Interrupt settings
27     //External
28     EICRA = 0b00001010; //INT0, INT1, falling edge at RD0(US) and 1
29     EIMSK = 0b00000011; //Enable INT0, enable INT1
30     //Internal
31     //Timer0 Counter T0 (PD4, Colour)
32     TCCR0A = 0b00000000; //Normal-Mode
33     TCCR0B = 0b00000110; //00doesn't matter,00unimpl.,110External clock sourc
34     TIMSK0 = 0b00000000; //Timer 0 no Interrupt
35
36     //Timer1 PWMA / B
37     TCCR1A = 0b10100001; //Clear OC1A/OC1B on Compare Match when upcounting.
38     TCCR1B = 0b00000010; //00doesn't matter,00unimpl.,0PWM,011TMR1-Pres1:8
39     TIMSK1 = 0b00000000; //Timer 1 no Interrupt
40
41     //Timer2 Time divider
42     TCCR2A = 0b00000000; //0000Normal port operation,00unimpl.,00WGM2[1:0]
43     TCCR2B = 0b00000010; //0WGM22,010Prescaler 1:8
44     TIMSK2 = 0b00000001; //Interrupt at TMR2-OVFL, 128us
45     //ADC settings
46     ADMUX = 0b01100001; //AVCC with external capacitor at AREF pin, left ad
  
```

Data Visualizer:

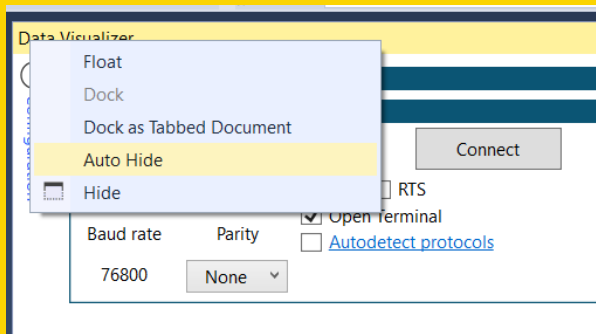
Baud rate muss geändert werden
von **9600** auf **76800**

Rechtsklick auf Data Visualizer

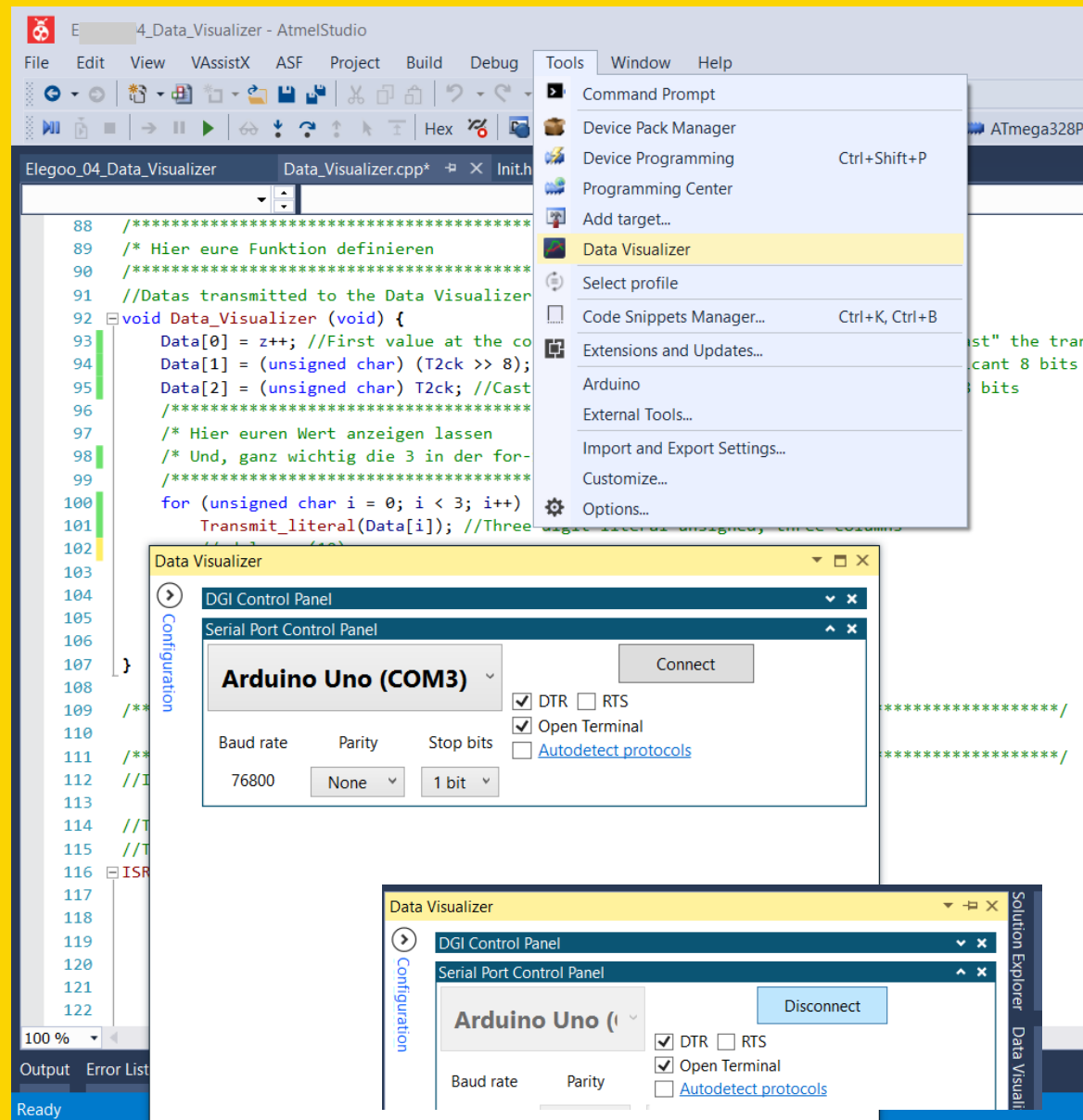


Klick Dock

Rechtsklick auf Data Visualizer



Klick Auto Hide



```

2.2 Data_Visualizer - Microchip Studio
File Edit View VAssistX ASF Project Build Debug Tools Window
Data_Visualizer.cpp Init.h USART_Functions.h
81 //Warten_auf_Starttaster(); //
82
83 /**Testschleife*****/
84 while(0) { //Einschalten -> 1, Ausschalten -> 0
85     Stop(); //Alle Motoren aus
86     Data_Visualizer(); //Sensorwerte anzeigen
87 }
88 /**Ende Testschleife*****/
89
90 while (1) { //Endlosschleife
91     Data_Visualizer();
92     _delay_ms(50);
93 }
94 }
95
96 /*******/
97 //Other functions
98
99 //Data being transmitted to the Data Visualizer
100 //Data[0] bis Data[7] werden auf LCD angezeigt
101 //Data[8] und Data[9] nur beim Data Visualizer
102 void Data_Visualizer (void) {
103     Data[0] = z++;
104     Data[1] = (unsigned char) (T2ck >> 8);
105     Data[2] = (unsigned char) T2ck;
106     Data[3] = Black_n_White;
107     Data[4] = (Encoder >> 8); //HI-Byte
108     Data[5] = Encoder; //LO-Byte
109     Data[6] = Start_Taster;
110     Data[7] = Abstand_2;
111
112 //Nur auf Data Visualizer angezeigt
113 Data[8] = MSB_Gyr_X_BN0055;
114 Data[9] = LSB_Gyr_X_BN0055;
115 }
116

```

2.000

HEX 7D0

DEC 2.000

OCT 3 720

BIN 0111 1101 0000

208

HEX D0

DEC 208

OCT 320

BIN 1101 0000

1.792

HEX 700

DEC 1.792

OCT 3 400

BIN 0111 0000 0000

7

HEX 7

DEC 7

OCT 7

BIN 0111

Advanced Mode Quick Launch (Ctrl+Q)

Visualizer

DGI Control Panel

Serial Port Control Panel

USB-SERIAL CH340 (COM9) Disconnect

DTR RTS

Open Terminal Autodetect protocols

Baud rate Parity Stop bits

76800 None 1 bit

Terminal 12

15	1	204	0	0	0	8	26	0	81
15	1	125	32	0	0	8	27	0	81
16	3	125	32	0	0	8	27	0	81
16	3	125	32	0	0	8	27	0	81
16	3	125	32	0	0	8	27	0	81
16	3	125	32	0	0	8	27	0	81
16	3	125	32	0	0	8	27	0	81
16	3	125	32	0	0	8	27	0	81
17	5	47	0	0	0	8	27	0	81
17	5	47	0	0	0	8	27	0	81
17	5	47	0	0	0	8	27	0	81
17	5	47	0	0	0	8	27	0	81
17	5	47	0	0	0	8	27	0	81
17	5	47	0	0	0	8	27	0	81
17	5	47	0	0	0	8	27	0	81
17	5	47	0	0	0	8	27	0	81
17	5	47	0	0	0	8	27	0	81
18	6	224	0	0	0	8	29	0	81
18	6	224	0	0	0	8	29	0	81
18	6	224	0	0	0	8	29	0	81
18	6	224	0	0	0	8	29	0	81
18	6	224	0	0	0	8	29	0	81

Clear Add \r\n Hexadecimal Values Show Timestamp

Automatically Scroll to End

Header einbinden

```
32  *A7 Ana ADC7 Dist-1
33  */
34
35  #define F_CPU 16000000UL //16MHz required for delay
36
37  #include <avr/io.h>
38  #include <util/delay.h> //Needed for using _delay_...
39  #include <avr/interrupt.h> //External and internal Interrupts
40  #include "Init.h"
41  #include "USART_Functions.h"
42  #include "Interrupt_Service_Routines.h"
43  #include "I2C.h"
44  #include "Drive_Functions.h"
45
```

Integer 0 - 65535

11001010 00110100 = 51764

```
65  #define PWMB OCR1B //Overflow compare register B
66
67
68  void Data_Visualizer (void); //Daten, die angezeigt werden sollen
69  void Warten_auf_Starttaster (void); //Nur Daten werden angezeigt
70
71  unsigned char z;
72
73  int main(void) {
74  Trinit(); //Initialisierung des Mikrocontrollers
```

Data Visualizer

```
99  //Data beeing transmitted to the Data Visualizer
100 //Data[0] bis Data[7] werden auf LCD angezeigt
101 //Data[8] und Data[9] nur beim Data Visualizer
102 void Data_Visualizer (void) {
103     Data[0] = z++;
104     Data[1] = (unsigned char) (T2ck >> 8);
105     Data[2] = (unsigned char) T2ck;
106     Data[3] = Black_n_White;
107     Data[4] = (Encoder >> 8); //HI-Byte
108     Data[5] = Encoder; //LO-Byte
109     Data[6] = Start_Taster;
110     Data[7] = Abstand_2;
111
112     //Nur auf Data Visualizer angezeigt
113     Data[8] = MSB_Gyr_X_BNO055;
114     Data[9] = LSB_Gyr_X_BNO055;
```

HEX	FFFF
DEC	65.535
OCT	177 777
BIN	1111 1111 1111 1111

Array (Feld):

Data[10]

10 Elemente mit gleichem Namen
aber unterschiedlichen Indizes.

Data[0], Data[1] ... Data[9]

Ein Array beginnt immer mit Element [0]

void Funktionsname (Variable)

Beim Aufruf der Funktion wird ein Wert mit übergeben an die Funktion.

Beispiel:

...

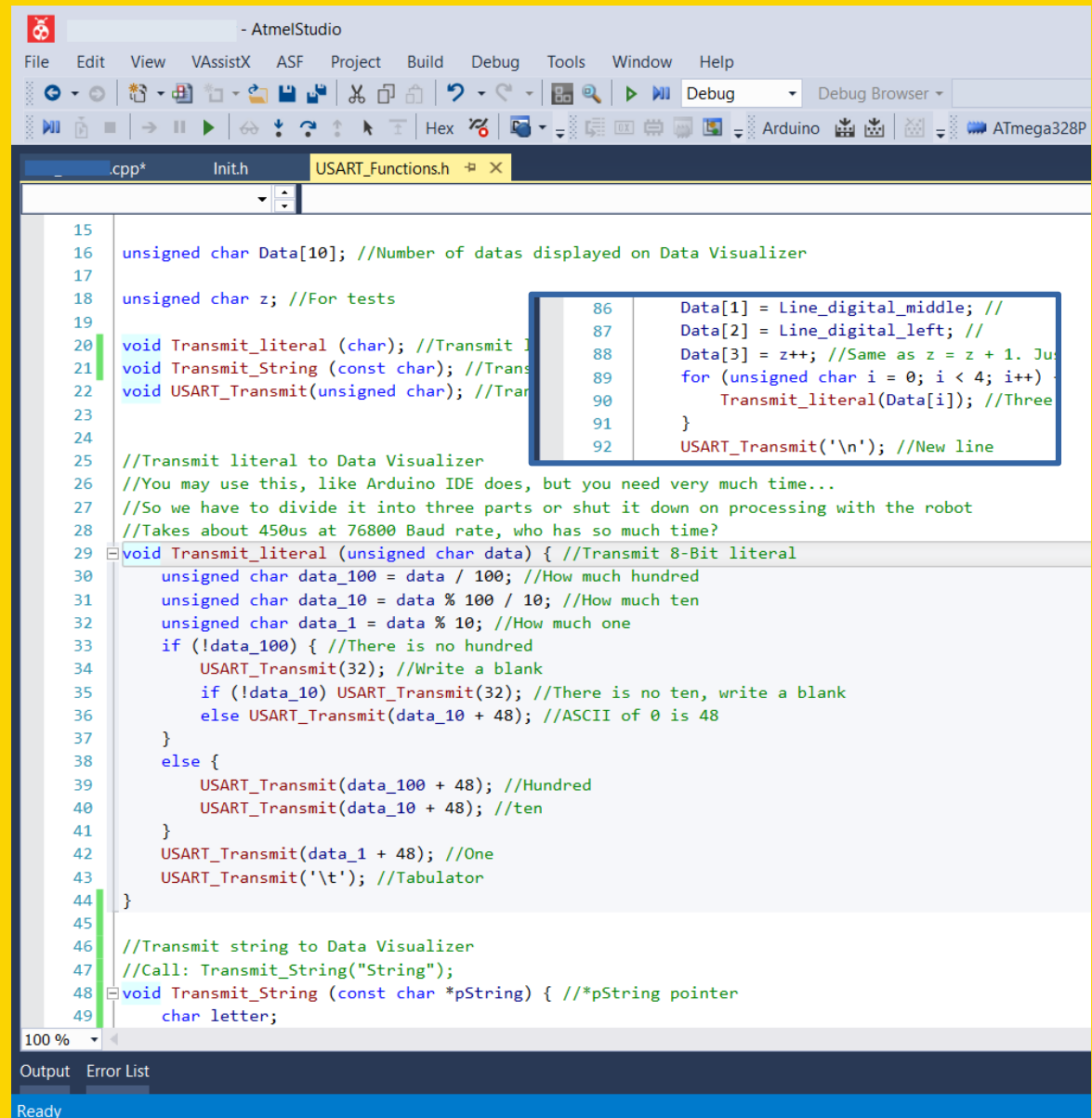
```
mal_drei_nehmen(8);
```

...

```
void mahl_drei_nehmen(Faktor) {  
    Produkt = 3 * Faktor;  
}
```

In Produkt ist jetzt der Wert 24 gespeichert.

Aufgabe: Implementiert (d.h.: baut in das bestehende Programm ein) eine Funktion mit der ihr Werte verdoppelt, die an die Funktion übergeben wird. Und zeigt sie anschließend an. Das alles soll im Hauptfenster (Data_Visualizer.cpp) geschehen. (Hilfe: Zuerst eine Variable deklarieren, dann im Deklarationsteil die Funktion bekannt machen, [denkt daran, dass ihr einen unsigned char übergeben wollt] Unterhalb der main die Funktion definieren, {} nicht vergessen. In der main aufrufen. Im Data_Visualizer ausgeben.) Schaut euch die Ergebnisse an. Was fällt euch auf?



```
- AtmelStudio  
File Edit View VAssistX ASF Project Build Debug Tools Window Help  
Debug Browser  
Arduino ATmega328P  
cpp* Init.h USART_Functions.h  
15 unsigned char Data[10]; //Number of datas displayed on Data Visualizer  
16  
17 unsigned char z; //For tests  
18  
19  
20 void Transmit_literal (char); //Transmit literal  
21 void Transmit_String (const char); //Transmit string  
22 void USART_Transmit(unsigned char); //Transmit 8-bit data  
23  
24  
25 //Transmit literal to Data Visualizer  
26 //You may use this, like Arduino IDE does, but you need very much time...  
27 //So we have to divide it into three parts or shut it down on processing with the robot  
28 //Takes about 450us at 76800 Baud rate, who has so much time?  
29 void Transmit_literal (unsigned char data) { //Transmit 8-Bit literal  
30     unsigned char data_100 = data / 100; //How much hundred  
31     unsigned char data_10 = data % 100 / 10; //How much ten  
32     unsigned char data_1 = data % 10; //How much one  
33     if (!data_100) { //There is no hundred  
34         USART_Transmit(32); //Write a blank  
35         if (!data_10) USART_Transmit(32); //There is no ten, write a blank  
36         else USART_Transmit(data_10 + 48); //ASCII of 0 is 48  
37     }  
38     else {  
39         USART_Transmit(data_100 + 48); //Hundred  
40         USART_Transmit(data_10 + 48); //ten  
41     }  
42     USART_Transmit(data_1 + 48); //One  
43     USART_Transmit('\t'); //Tabulator  
44 }  
45  
46 //Transmit string to Data Visualizer  
47 //Call: Transmit_String("String");  
48 void Transmit_String (const char *pString) { //pString pointer  
49     char letter;  
86     Data[1] = Line_digital_middle; //  
87     Data[2] = Line_digital_left; //  
88     Data[3] = z++; //Same as z = z + 1. Just increment z  
89     for (unsigned char i = 0; i < 4; i++)  
90         Transmit_literal(Data[i]); //Three times  
91     }  
92     USART_Transmit('\n'); //New line
```

**Erst
Aufgabe
lösen,
dann
weiter
schauen!**



American Standard Code for Information Interchange (ASCII)

```
04_Data_Visualizer - AtmelStudio
File Edit View VAssistX ASF Project Build Debug Tools Window Help
Debug Debug Browser led
Data_Visualizer.cpp* Inith USART_Functions.h
13
14 //unsigned char z; //For tests
15
16 void Transmit_literal (char); //Transmit literal to Data Visualizer Monitor
17 void Transmit_String (const char); //Transmit letters to Data Visualizer Monitor
18 void USART_Transmit(unsigned char); //Transmit ASCII to Data Visualizer
19
20
21 //Transmit literal to Data Visualizer
22 //You may use this, like Arduino IDE does, but you need very much time...
23 //So we have to divide it into three parts or shut it down on processing with the robot
24 //Takes about 450us at 76800 Baud rate, who has so much time?
25 void Transmit_literal (unsigned char data) { //Transmit 8-Bit literal
26     unsigned char data_100 = data / 100; //How much hundred
27     unsigned char data_10 = data % 100 / 10; //How much ten
28     unsigned char data_1 = data % 10; //How much one
29     if (!data_100) { //There is no hundred
30         USART_Transmit(32); //Write a blank
31         if (!data_10) USART_Transmit(32); //There is no ten, write a blank
32         else USART_Transmit(data_10 + 48); //ASCII of 0 is 48
33     }
34     else {
35         USART_Transmit(data_100 + 48); //Hundred
36         USART_Transmit(data_10 + 48); //ten
37     }
38     USART_Transmit(data_1 + 48); //One
39     USART_Transmit('\t'); //Tabulator...
40     //USART_Transmit(9); //...or ASCII for horizontal tab, HT
41 }
42
43 //Transmit string to Data Visualizer
44 //Call: Transmit_String("String");
45 void Transmit_String (const char *pString) { // *pString pointer
46     char letter; //Local variable
47     letter = *pString; // The content of the registers, the pointer is

```

Die Großbuchstaben fangen bei 65 an. A → 65ASCII.

Die Kleinbuchstaben fangen bei 97 an. a → 97ASCII.

Die Ziffern fangen bei 48 an, 0 → 48ASCII.

Dez	Hex	Okt	ASCII	Dez	Hex	Okt	ASCII	Dez	Hex	Okt	ASCII	Dez	Hex	Okt	ASCII
0	00	000	NUL	32	20	040	SP	64	40	100	@	96	60	140	`
1	01	001	SOH	33	21	041	!	65	41	101	A	97	61	141	a
2	02	002	STX	34	22	042	"	66	42	102	B	98	62	142	b
3	03	003	ETX	35	23	043	#	67	43	103	C	99	63	143	c
4	04	004	EOF	36	24	044	\$	68	44	104	D	100	64	144	d
5	05	005	ENQ	37	25	045	%	69	45	105	E	101	65	145	e
6	06	006	ACK	38	26	046	&	70	46	106	F	102	66	146	f
7	07	007	BEL	39	27	047	'	71	47	107	G	103	67	147	g
8	08	010	BS	40	28	050	(72	48	110	H	104	68	150	h
9	09	011	HT	41	29	051)	73	49	111	I	105	69	151	i
10	0A	012	LF	42	2A	052	*	74	4A	112	J	106	6A	152	j
11	0B	013	VT	43	2B	053	+	75	4B	113	K	107	6B	153	k
12	0C	014	FF	44	2C	054	,	76	4C	114	L	108	6C	154	l
13	0D	015	CR	45	2D	055	-	77	4D	115	M	109	6D	155	m
14	0E	016	SO	46	2E	056	.	78	4E	116	N	110	6E	156	n
15	0F	017	SI	47	2F	057	/	79	4F	117	O	111	6F	157	o
16	10	020	BIF	48	30	060	0	80	50	120	P	112	70	160	p
17	11	021	DC1	49	31	061	1	81	51	121	Q	113	71	161	q
18	12	022	DC2	50	32	062	2	82	52	122	R	114	72	162	r
19	13	023	DC3	51	33	063	3	83	53	123	S	115	73	163	s
20	14	024	DC4	52	34	064	4	84	54	124	T	116	74	164	t
21	15	025	NAK	53	35	065	5	85	55	125	U	117	75	165	u
22	16	026	SYN	54	36	066	6	86	56	126	V	118	76	166	v
23	17	027	ETB	55	37	067	7	87	57	127	W	119	77	167	w
24	18	030	CAN	56	38	070	8	88	58	130	X	120	78	170	x
25	19	031	EM	57	39	071	9	89	59	131	Y	121	79	171	y
26	1A	032	SUB	58	3A	072	:	90	5A	132	Z	122	7A	172	z
27	1B	033	ESC	59	3B	073	;	91	5B	133	[123	7B	173	{
28	1C	034	FS	60	3C	074	<	92	5C	134	\	124	7C	174	
29	1D	035	GS	61	3D	075	=	93	5D	135]	125	7D	175	}
30	1E	036	RS	62	3E	076	>	94	5E	136	^	126	7E	176	~
31	1F	037	US	63	3F	077	?	95	5F	137	_	127	7F	177	DEL

```
67
68 //Main routine
69 int main(void) {
70     Init(); //Initialize the MC
71     USART_Init(MYUBRR); //Transmission/reception initialize
72     sei(); //Enable all interrupts
73     cli(); //Clear all interrupts
74     while (1) {
75         Transmit_String("Text: May be as long as you want, but lasts and lasts and...! \n");
76         Data_Visualizer(); //Define the values to be displayed and transmit them
77         /* Hier eure Funktion aufrufen */
78         /* Hier eure Funktion definieren */
79     }
80 }
81
82 /* Hier eure Funktion definieren */
83 //Other functions
84 /* Hier eure Funktion definieren */
85 /* Hier eure Funktion definieren */
86 //Datas transmitted to the Data Visualizer
87 void Data_Visualizer (void) {
88     Data[0] = z++; //First value at the console. Same as z = z + 1. Just to see how "fast" the transmission is
89     Data[1] = (unsigned char) (T2ck >> 8); //Casting integer to character, most significant 8 bits
90     Data[2] = (unsigned char) T2ck; //Casting integer to character, least significant 8 bits
91     /* Hier euren Wert anzeigen lassen */
92     /* Und, ganz wichtig die 3 in der for-Schleife durch 4 ersetzen!!! */
93     for (unsigned char i = 0; i < 3; i++) {
94         Transmit_literal(Data[i]); //Three digit literal unsigned, three columns
95         // _delay_ms(10);
96     }
97     USART_Transmit('\n'); //New line
98     //USART_Transmit(10); //ASCII for "line feed", LF
99
100
101
```

Data Visualizer

Configuration

DGI Control Panel

Serial Port Control Panel

Arduino Uno (Disconnect

DTR RTS

Open Terminal

Autodetect protocols

Baud rate Parity

76800 None

Terminal 33

Text: May be as long as you want, but lasts and lasts and...!
135 0 242

Text: May be as long as you want, but lasts and lasts and...!
136 1 63

Text: May be as long as you want, but lasts and lasts and...!
137 1 140

Text: May be as long as you want, but lasts and lasts and...!
138 1 217

Text: May be as long as you want, but lasts and lasts and...!
139 2 39

Text: May be as long as you want, but lasts and lasts and...!
140 2 116

Text: May be as long as you want, but lasts and lasts and...!
141 2 193

Text: May be as long as you want, but lasts and lasts and...!
142 3 14

Text: May be as l

Clear Add \r\n Hexadecimal Values

Show Timestamp Automatically Scroll to End

Musterlösung:

//Deklaration der Variablen in der der doppelte Wert gespeichert wird.
unsigned char Doppelt;

//Deklaration der Funktion
void Verdopplung (unsigned char);

//Aufruf der Funktion in der main

...
 Verdoppelung(z);

...

//Definition der Funktion
void Verdoppelung (unsigned char Faktor) {
 Doppelt = Faktor * 2;
}

//Anzeigen des Wertes

...
 Data[3] = Doppelt;

...

```
04_Data_Visualizer - AtmelStudio
File Edit View VAssistX ASF Project Build Debug Tools Window Help
Debug Browser
Data_Visualizer.cpp x Init.h USART_Functions.h
Data_Visualizer void Data_Visualizer(void)
50 /*****
51  /* Hier eure Variable (unsigned char) deklarieren */
52  /*****
53
54  //unsigned char Data[10]; //Number of datas displayed on Data Visualizer
55
56  //Funktionen, declaration
57  void Init (void); //Initialize Microcontroller
58  //void USART_Transmit(unsigned char); //Transmit ASCII to Data Visualizer
59  void USART_Init(unsigned int); //Initialize USART Transmission/Reception
60  //void Transmit_literal (char); //Transmit literal to Data Visualizer Monitor
61  //void Transmit_String (const char); //Transmit letters to Data Visualizer Monitor
62  void Data_Visualizer (void); //Define datas to be visualized
63  /*****
64  /* Hier eure Funktion deklarieren */
65  /*****
100 %
76 Data_Visualizer(); //Define the values to be displayed and transmit them
77 /*****
78  /* Hier eure Funktion aufrufen */
79  /*****
80  }
81  }
82  /*****
83  //Other functions
84  /*****
85  /* Hier eure Funktion definieren */
86  /*****
87  //Datas transmitted to the Data Visualizer
88  void Data_Visualizer (void) {
89  Data[0] = z++; //First value at the console. Same as z = z + 1. Just to see how
90  Data[1] = (unsigned char) (T2ck >> 8); //Casting integer to character, most sig
91  Data[2] = (unsigned char) T2ck; //Casting integer to character, least significa
92  /*****
93  /* Hier euren Wert anzeigen lassen */
94  /* Und, ganz wichtig die 3 in der for-Schleife durch 4 ersetzen!!! */
95  /*****
96  for (unsigned char i = 0; i < 3; i++) {
Output Error List
Ready
```

Montage und Programmierung
eines Roboters für
ROBOCUP JUNIOR RESCUE
mit Arduino Nano
Teil 2.4: Black and White