

Einführung in die Mikrocontrollerprogrammierung

17. 2. 2014

Wir gehen genau so vor wie im Praktikum. Auf gleiche Weise arbeitet sich auch der Profi in eine neue Mikrocontrollerfamilie ein (geht nicht von heute auf morgen...).

Die Ausrüstung:

- Eine Entwicklungssoftware aus dem Internet (kostet nichts).
- Ein Starterkit (kostet einige zehn bis wenige hundert Euro).
- Bei Bedarf etwas zusätzliche Hardware zum Spielen.

Wir verwenden:

- das Entwicklungssystem AVR Studio 4 (Atmel) mit dem GNU C-Compiler,
- das Starterkit Atmel STK 500,
- der XY-Adapter 09A (Eigenbau),
- ein Oszilloskop, das wir wie einen Plotter betreiben (XY-Darstellung), so daß wir mit dem Elektronenstrahl auf dem Bildschirm zeichnen können.

Unser Ziel: Auf dem Bildschirm etwas darstellen, das nach was aussieht.

Das Programm ist schon vorgefertigt. Sie müssen nur die Funktionen so einsetzen, daß die gewünschte Darstellung zustande kommt.

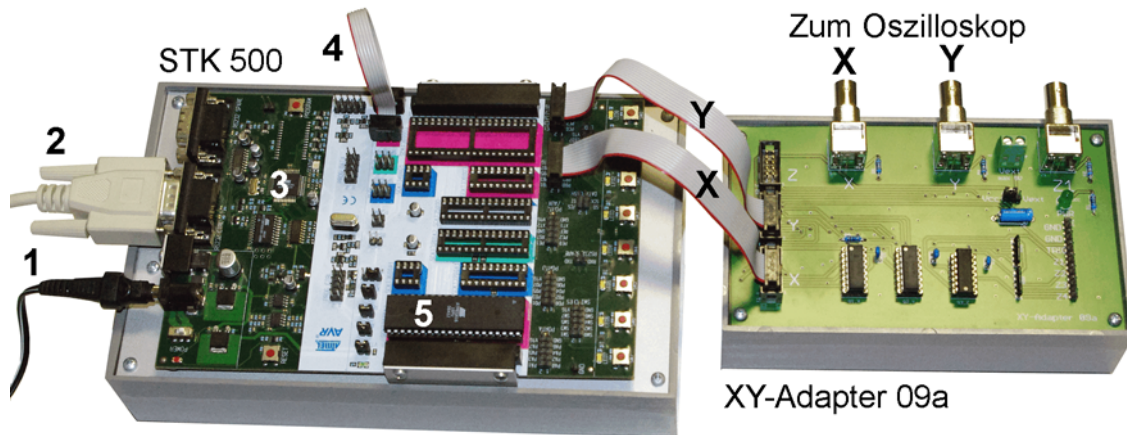
Jedes Vorhaben ist ein Projekt. Wenn Sie etwas Neues beginnen wollen, müssen Sie im Entwicklungssystem eine neues Projekt einrichten. Es sind ein paar Tastenanschläge und Mausklicks mehr. Das haben wir hier bereits getan.

Der Entwicklungsgang:

1. Den Programmtext eintippen.
2. Das Programm übersetzen lassen (BUILD). Ggf. Fehler beseitigen
3. Das Programm in den Mikrocontroller laden.
4. Zusehen, ob es funktioniert. Wenn nicht, überlegen, woran es liegen könnte. Dann zurück zu Schritt 1.

Alle Schritte so oft durchlaufen, bis es funktioniert...

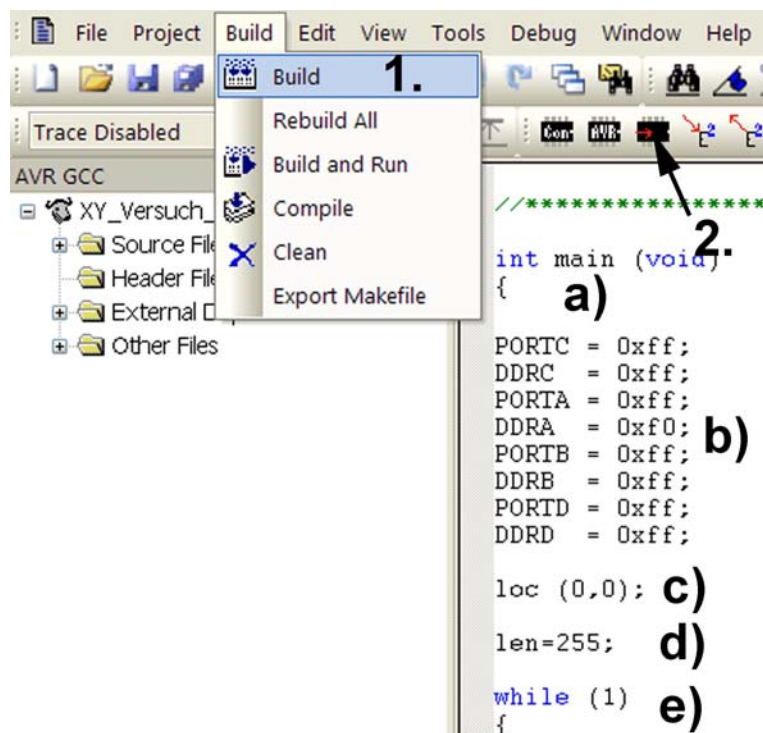
Die Ausrüstung:



1 - Stromversorgung; 2 - Programmierdaten vom PC; 3 - Programmer; 4 - internes Programmierkabel; 5 - Mikrocontroller.

Die wichtigsten Mausklicks, damit es läuft:

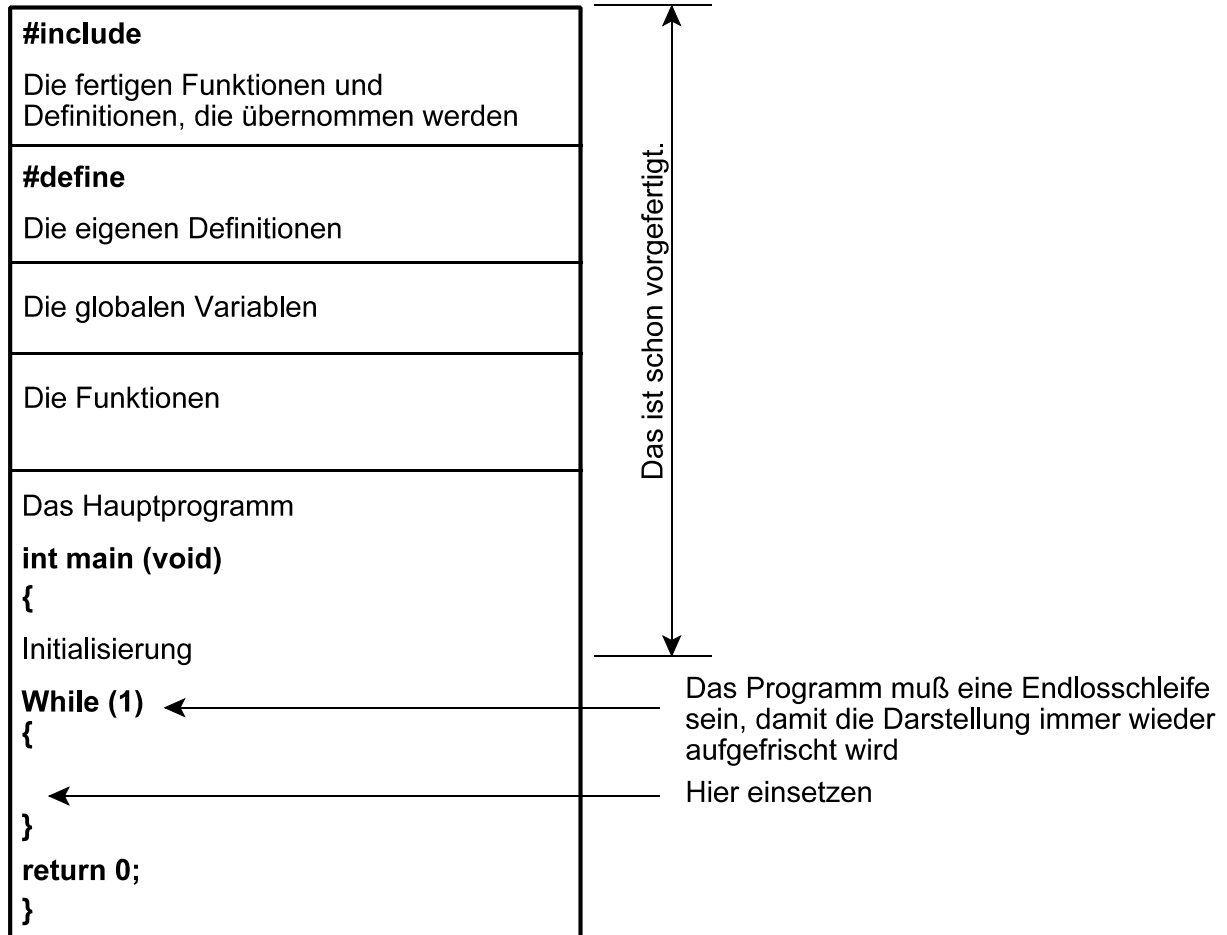
1. Übersetzen.
2. Das Programm in den Mikrocontroller laden.



- a) Beginn des Hauptprogramms
- b) Die Deklaration der E-A-Ports
- c) Positionierung auf den Koordinatenursprung (0, 0)
- d) Länge = 255, um das umschreibende Quadrat zu zeichnen
- e) Beginn der Endlosschleife

Wir verwenden die Programmiersprache C. Sie ist alt und von Grund auf häßlich. Es ist aber die am meisten verwendete Programmiersprache.

Das C-Programm

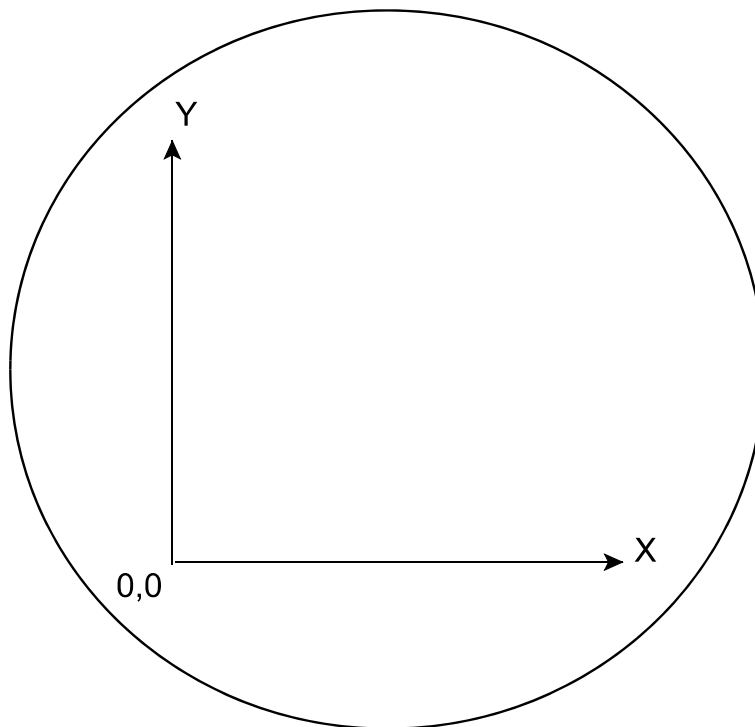


Ihre Programmieraufgabe besteht eigentlich nur darin, eine Folge von Funktionen einzugeben, die die gewünschte Darstellung erzeugt.

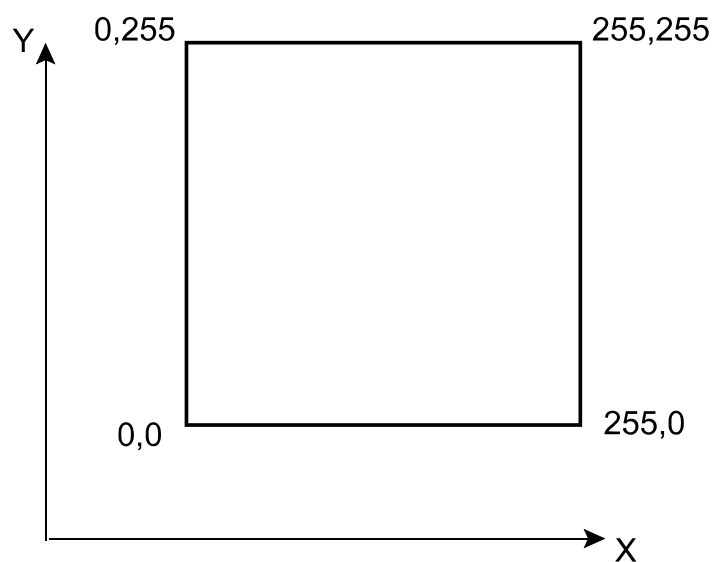
Achtung – nach jeder Anweisung kommt ein Semikolon.

Ein Quadrat, das den gesamten Darstellbereich umschreibt, ist schon vorgefertigt. Versuchen Sie zunächst einmal, eine Diagonale von links unten nach rechts oben zu ziehen. Dann können Sie probieren, beispielsweise eine Ziffer oder einen Buchstaben darzustellen.

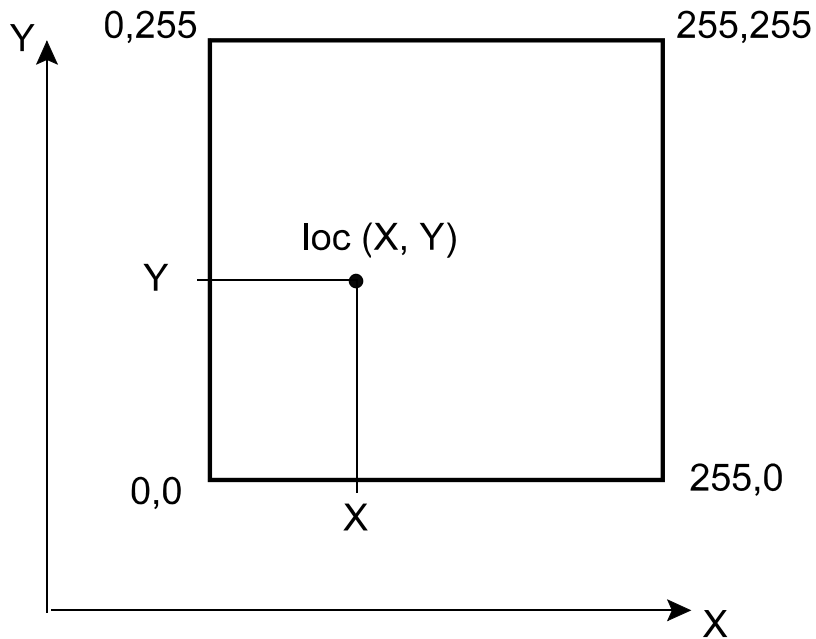
Das Prinzip der Darstellung auf dem Oszilloskop:



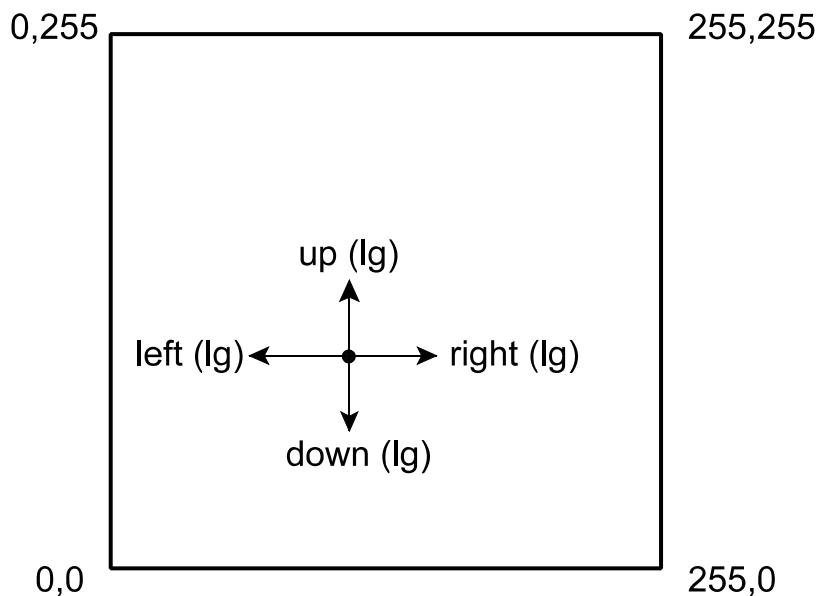
Das sind die Grenzen des Darstellbereichs. Die Darstellung besteht aus Bildpunkten (Pixeln). In jeder Koordinatenachse sind es maximal 256 Pixel. Sie werden jeweils von 0 bis 255 gezählt.



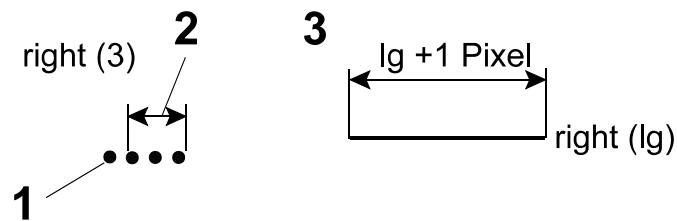
Mit der Funktion **loc** (X,Y) lenken Sie den Elektronenstrahl auf eine bestimmte Position, so daß dort ein Bildpunkt dargestellt wird. (**loc** steht für das englische *locate*.)



Senkrechte und waagerechte Linien zeichnen Sie mit diesen vier Funktionen:



Der Funktionsparameter *lg* gibt an, wieviele Pixel jeweils neu zu zeichnen sind. Er ist also ein Maß für die Länge der Linie. Eine so gezeichnete Linie ist $lg + 1$ Pixel lang. Das läßt sich einfach anhand eines Beispiels erklären:



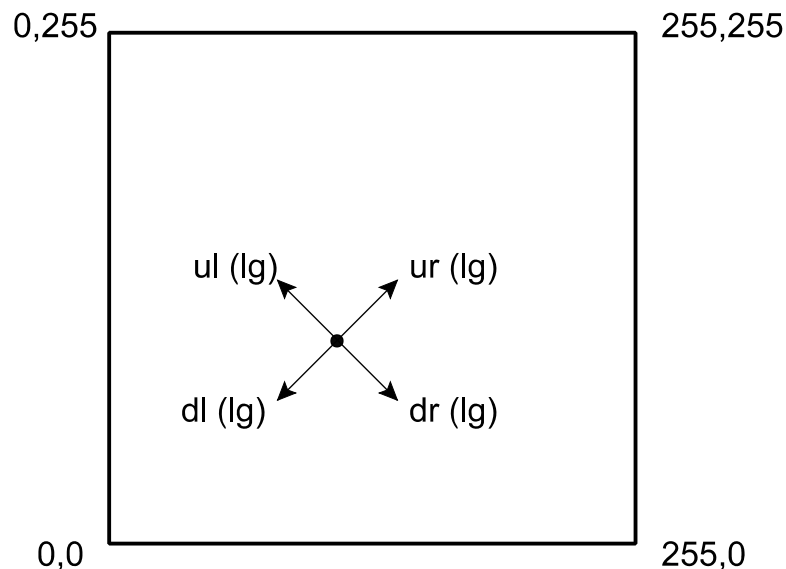
1 - der Ausgangspunkt bzw. das ursprüngliche Pixel (als Ergebnis einer vorhergehenden Zeichenfunktion oder einer Funktion **loc** (X, Y). 2 - eine Funktion **right** (3) zeichnet drei weitere Pixel. Die Linie ist also insgesamt vier Pixel lang.

Buchstaben, Ziffern usw. werden aus Linienstücken zusammengesetzt. Oftmals sind diese Linienstücke gleich lang. Die Länge soll aber steuerbar sein, beispielsweise um die Schriftgröße einstellen zu können. Hierfür haben wir eine Variable **len** vorgesehen. Wird eine Linienzeichenfunktion mit einem Parameter Null aufgerufen (z. B. **right** (0)), so wird eine Linie der Länge **len** gezeichnet. Beispiel:

```
len = 3;
up (len); // Es wird eine Linie senkrecht nach oben gezeichnet, die
           // insgesamt 4 Pixel lang ist.
```

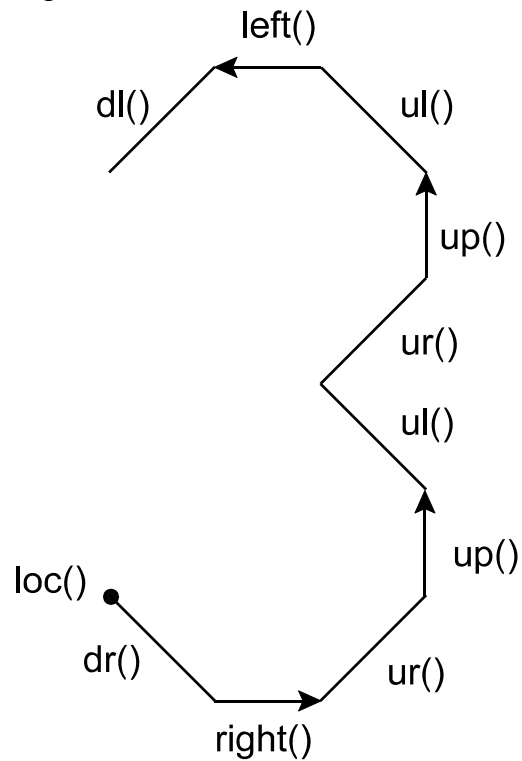
Das gilt sinngemäß auch für die nachfolgend beschriebenen Funktionen.

Mit diesen vier Funktionen können Sie schräge Linien zeichnen, die um 45° geneigt sind:



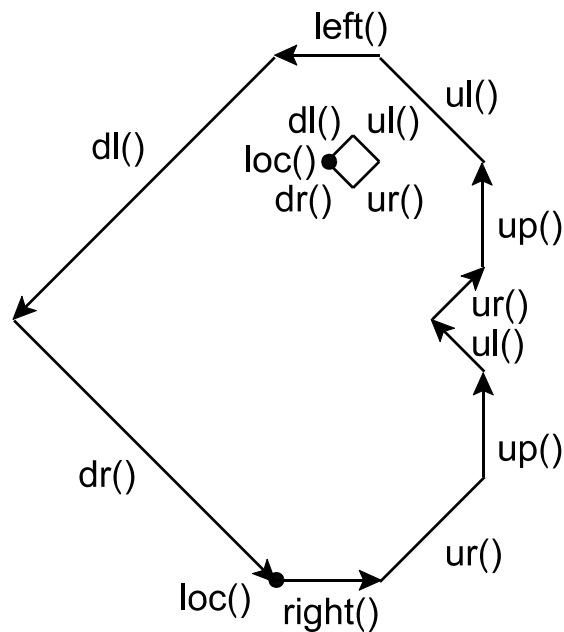
Alle Bezeichnungen sind aus dem Englischen abgeleitet. Schwierig sollte es eigentlich nicht sein...

So wird die Ziffer Drei dargestellt:



In die Klammern passende Werte einsetzen. Z. B. **loc** (30,30) und Nullen in die Lineinzeichenfunktionen (**dr** (0); **right** (0) usw.) Die Längenzuweisung im fertigen Text z. B. auf **len** = 9 einstellen. Die Drei besteht dann aus Linienstücken, von denen jedes 10 Pixel lang ist

Das (vereinfachte) Pacman-Symbol ist schon etwas anspruchsvoller...



Viel Erfolg!

Wer fleißig programmiert, kann es unter anderem auch dazu bringen:

