MCK PonyProg

PonyProg2000.gif

Bei der Verwendung eines USB- Programmers kann die Programmierung des Mikrocontrollers bequem aus BASCOM heraus durchgeführt werden. Da die Anzahl der angebotenen USB- Programmer aber eine unüberschaubare Vielfalt angenommen hat und jeder Programmer so seine Eigenheiten hat, bin ich hier noch einmal einen anderen Weg gegangen. Alle Beispiele lassen sich selbstverständlich (und auch einfacher) mit einem USB- Programmer durchführen und unsere Experimentierplatine hat dafür natürlich auch einen Anschluß. Da der Kurs aber vordergründig für Schüler mit normalerweise meist schmalem Budget gedacht ist, befindet sich auf unserer ersten Experimetierplatine auch ein serieller Programmer. Auch wenn serielle Anschlüsse an heutigen Rechnern immer seltener werden, so stehen in den Schulen und Jugendzimmern doch sehr häufig noch Rechner, welche genau diese Buchsen zur Verfügung stellen.

Aus diesem Grund folgt hier einmalig eine Erklärung zum Umgang mit Ponyprog, einem kostenlosen Tool für die serielle Programmierung unserer Atmelcontroller. Im Kursus selber wird auf Ponyprog dann nicht mehr ausführlich eingegangen.

Wie erwähnt ist auch Ponyprog ein Programm, welches wir kostenlos downladen können. Hier der Link zur Seite:
Link zum PonyProg Download

Auch PonyProg müssen wir für unsere Zwecke erst vorbereiten. Als erstes muß das Programm kalibriert werden. Wenn wir PonyProg gestartet haben klicken wir also zunächst auf "Setup" und anschließend auf "Calibration". Folgt hier den Anweisungen zur Kalibrierung!

Als Zweites müssen wir PonyProg mitteilen, wo sich unser Programmieradapter befindet. Dazu öffenen wir folgendes Menue in gleicher Weise, wie wir das bei BASCOM getan haben. Auch hier hilft wieder eine Grafik, welche der Bildschirmansicht von PonyProg nach dem Anklicken von "Setup" und "Interface Setup" entspricht:

Pony setup.png

PonyProg kann mit Programmern am parallelen Port (Druckerport) oder aber am seriellen Port betrieben werden. Wir stellen hier natürlich unseren seriellen Port ein. Darunter muss noch angegebenwerden, welche Portnummer verwendet werden soll. In den meisten Fällen wird dies COM1 sein, sollten zwei serielle Ports vorhanden sein, ist auch COM2 möglich. Im Zweifelsfall können wir unter Windows=> System => Gerätemanager nachsehen, welcher Port aktuell verfügbar ist.

Um jetzt PonyProg zum "Flashen" unseres Atmel Mikrocontrollers verwenden zu können müssen wir allerdings einen kleinen Umweg gehen. Wir benötigen von BASCOM unser Programm in hexadezimaler Form, also im sogenannten "HEX-Format", damit es in Ponyprog weiter verarbeitet werden kann. Dies ist allerdings kein Problem, da BASCOM beim Compilieren auch die gesuchte Hex-Datei erzeugt. Unter Compilieren versteht man die Umwandlung des in BASIC geschriebenen Programms in andere Formate, wie beispielsweise das Hex- Format.

Wichtig ist, dass Ihr Euch merkt, in welchem Ordner Ihr Euer BASCOM- Programm, welches Ihr auf den Controller übertragen wollt abgespeichert hat, denn BASCOM legt die compilierten Dateien in demselben Ordner ab, in dem auch das eigentliche Programm gespeichert wurde.

Mit steigender Anzahl von BASCOM- Programmen werden die compilierten Dateien schnell mehr und man verliert leicht den Überblick. Ihr tut gut daran Euch von Anfang an anzugewöhnen Eure BASCOM-Programme möglicht projektbezogen in einem jeweils neuen Ordner abzulegen. Das erspart Euch in Zukunft viel Sucharbeit, da Ihr bereits am Ordnernamen erkennen könnt, welches Programm im "Inneren"compiliert abgelegt ist.

Nach Einrichten der Grundeinstellungen könnt Ihr jetzt auf einfache Weise PonyProg zum Flashen Eurer Controller einsetzen.
Zum Abschluß dieser Einleitung zeige ich Euch in gewohnter Form noch wie die Einzelschritte zum erfolgreichen Flashen aussehen. Alle weiteren Funktionen von Ponyprog lassen wir zunächst unbeachtet.Diese werden zu einem späteren Zeitpunkz ergänzt, wenn wir diese für unsere Arbeit benötigen.

Wie in anderen Windows- Programmen auch, klickt Ihr zuerst auf den Button "Datei öffnen" (1). Die Bildschirmanzeige hierzu habe ich nicht extra aufgeführt, da diese ja bei jedem von Euch anders ist. Wie bei anderen Programmen gewohnt, wählt Ihr hier den Ordner aus, in dem das gewünschte Programm gespeichert ist. Mit einem Doppelklick wählt Ihr das gewünschte Programm mit der Endung: ".hex"aus.
Euer Programm erscheint dann ähnlich wie im Screenshoot mit diesen merkwürdigen Zweiergruppen, welche bei näherem Hinsehen nichts anderes als die erwähnten hexadezimalen Werte sind (Zur Erinnerung hexadezimale Darstellung mit den Zahlen 0-9 und den Buchstaben A-F).
Diese interessieren uns aber nicht näher. Viel wichtiger ist die Einstellung beim gelben Pfeil (2). Hier muß im linken Feld "AVR mikro" stehen und im rechten Feld "AVR auto". Im rechten Feld kannauch Euer Controller direkt bezeichnet sein, in diesem Fall also "Attiny 13". Sollte es unerwarteter Weise Schwierigkeiten mit der Auswahl "AVR auto" geben, wählt im rechten Tab einfach den verwendeten Controller direkt aus.
Als dritten Schritt klickt nun auf den Button auf den der grüne Pfeil (3) zeigt. Dieser bewirkt, dass das Programm zum Mikrocontroller übertragen wird, aber nicht ohne die Dialogbox, die nun eingeblendet wird. Dort klickt Ihr noch einmal auf "yes" (4), damit der Controller nicht unbeabsichtigt überschrieben wird, denn alles, was bisher im Chip gespeichert war ist mit dem Überschreiben unwiederbringbar verloren.

Ponyprog2.png