

## **Projektbeschreibung**

Bei der vorgestellten R8-Tiny-Anwendung handelt es sich in erster Priorität um ein Messgerät zur Aufzeichnung eines Bewegungsablaufs im industriellen Umfeld einer Giesserei.

Unterschiedliche Füllhöhen der Formkasten und Kaltlauf von Gussteilen an einer vollautomatischen Giessanlage waren Motivation dafür, die Füllhöhen elektronisch zu messen, um die Giesskurvenstabilität nachweisen zu können um den Ursachen der unerwünschten Abweichungen auf die Spur zu kommen.

Ablaufstörungen, unterschiedliche statische Giesshöhen über dem Ausfluss sowie Einflussnahme der Mitarbeiter auf die Vorgabegiesszeit und –kurven sind eine kleine Aufzählung möglicher Störeinflüsse.

Erschwerend kommt hinzu, dass die Giesskurve für das gleiche Modell scheinbar öfters neu mittels eines Masterhebels eingelesen werden muss, was einer neuen, veränderten Giesskurve gleichkommt. Nur gut geübte Giesser können die Giesskurve nach wenigen Versuchen optimal vorgeben.

Aus einer Giesskurve lassen sich verschiedene Kennzahlen ableiten, wie z.B. die Giesszeit, der maximale Giesshub, der mittlere Giesshub und auch das Ausflussgewicht. Diese Ergebnisse können dann kastenbezogen grafisch dargestellt werden und erlauben hierdurch Rückschlüsse über die Gleichmäßigkeit der Giesskurve und der abgegossenen Serie.

In zweiter Priorität liegt es nahe, das Gerät auch zur Betriebsdatenerfassung zu nutzen, da sich die Giesstakte und Ablaufstörungen sehr gut in einer kastenbezogenen Grafik darstellen lassen. Hieraus ergeben sich ebenfalls qualitätsrelevante Rückschlüsse.

Die Beigabe des Mikrokontrollers in der Dezemberausgabe 2005 waren für mich Anlass dafür, zu prüfen, ob mit dieser MPU mit mehreren analogen Schnittstellen u.a. die Giesskurvendaten erfasst, die Daten ausreichend schnell mittels der integrierten seriellen Schnittstelle via Visual-Basic an einen handelsüblichen Rechner übergeben werden können und die zur Verfügung stehende Zeit ausreicht, das ganze auch grafisch darzustellen.

Wie sich zeigte, ging das problemlos. D.h., man kann 2 Analogkanäle 20 mal pro Sekunde auslesen, was Aufgabe des Visual-Basic-Timers ist (alle 50 ms eine Abfrage). Der Timer wertet die Giesskurve aus und erledigt dann die entsprechenden Skalierungen für die diversen grafischen und textorientierten Anzeigen.

Mittels eines programmierten Messprogrammes liess sich feststellen, dass es möglich ist, weit mehr Messungen als nur 2 mal 20 Messungen pro Sekunde durchzuführen. In dem zur Verfügung stehenden Zeitintervall werden nun abwechselnd 5 Messungen pro Kanal durchgeführt und es werden dann die Mittelwerte aus 5 Messungen seriell übertragen (d.h., es werden 2 mal 100 Analogmessungen pro Sekunde durchgeführt).

In Kenntnis, dass Windows ein Multitasking-Betriebssystem ist, was eine Echtzeitfähigkeit ausschliesst, waren Anlass dazu, das Programm so zu ändern, dass ausser den Analogdaten auch die Zeit- und Taktdaten auf der MPU berechnet und somit auch diese Daten seriell übertragen werden. Zusätzlich werden nun auch „Altdaten“ gesendet, so dass das Visual-Basic-Programm prüfen kann, ob Daten fehlen. Sollten Daten fehlen, kann Visual-Basic diese nun mit berücksichtigen, da sie ja zur Verfügung stehen. Durch diese Methode gelang es, die Deckungsgleichheit aufeinanderfolgender Giesskurven weiter zu verbessern.

Integriert und zum Teil auch getestet ist die Verwendung eines Schrittmotors zum Betätigen des Giesshebels mittels einer speziellen Vorrichtung. Sollte dies gelingen (Drehmoment und Drehzahl des Schrittmotors müssen stimmen), ist es möglich, eine Giesskurve durch Steuern des Giesshebels „syntetisch“ zu erzeugen, das Ergebnis zu messen und gegebenenfalls regulierend einzugreifen, um die „Sollgiesskurve“ bzw. das „Sollausflussvolumen“ zu erreichen.

Das Gerät soll zu einem Stand-Alone-Messgerät mit einer LCD-Anzeige ausgerüstet werden. Die wichtigsten Giessparameter sollen dann grafisch auf dem Display wiedergegeben werden. Das sind im Wesentlichen die Giesskurve selbst, die kastenbezogenen Anzeigen betreffs Giesszeit, maximaler Giesshub, mittlerer Giesshub, Ausflussgewicht, Schwimmverhalten (Kanal 2), zeitbezogene Kastenleistung u.a.

Die Daten sollen im Gerät gespeichert werden (Datenloggerfunktion), so dass sie am Folgegiessstag per serieller Schnittstelle ausgelesen werden können (die Lösung fehlt noch).

Mit PC und angeschlossener Webcam ist es möglich, die abgegossenen Kasten als Bild aufzunehmen. Aus dem Bild lässt sich ein Streifenbild extrahieren und kastenbezogen in einer Grafik darstellen, was einen wunderbaren

Überblick über die Füllhöhenkonstanz während der Serie zulässt. Die Pixelfarben des Bildstreifens können ebenfalls ausgewertet werden, so dass aus dem Durchmesser des Giesstrichters die effektive Giesshöhe abgeleitet werden kann. Wünschenswert wäre es, wenn es mir gelingen würde, mit Hilfe eines Mikrokontrollers das Bildsignal eines CCD-Kameramoduls auslesen zu können um auch mit einem Stand-Alone-Messgerät nicht auf diese Möglichkeit verzichten zu müssen.

### **Programmkurzbeschreibung**

Das Hauptprogramm „main“ initialisiert zuerst den Systemtakt, die serielle Schnittstelle, die Timer und die verwendeten Ports. Danach verweilt der Programmablauf in der „while“-Schleife um sofort beim Empfang eines Steuercodes (rx\_selection) via dem Unterprogramm „select“ zur angedachten Aufgabe temporär zu verzweigen („mode“ muss hierzu „0“ gesetzt sein). Unter „mode = 0“ lassen sich somit 255 Aufgaben definieren.

Um auch Datenübertragungen zur MPU zu ermöglichen, wird mittels dem Steuercode „mode > 0“ gesetzt. Für jedes „mode“ kann eine spezielle Datenübertragung gewählt werden. Wichtig ist nur, dass nach Abschluss einer Modelfunktion „mode“ wieder auf „0“ gesetzt wird. Nur dann ist es wieder möglich, auf das Unterprogramm „select“ zugreifen zu können.

Nur die Timer-X und Timer-Y unterbrechen per Interrupt das Hauptprogramm „main“. Timer-Y ist mit einem festen Takt eingestellt, der auf 200 mal pro Sekunde eingestellt ist (200 Wandlungen pro Sekunde). Dieser Timer startet abwechselnd die beiden verwendeten AD-Kanäle „ad1“ und „ad2“ und liest sie auch aus („ad\_in2(x)“). Aus jeweils 5 Messungen je Kanal wird der Mittelwert berechnet und als Übergabewert gespeichert („ad1“ und „ad2“). Die Frequenz ist so gewählt, dass es im Wandler-Unterprogramm zu keiner wandlungsbedingten Wartezeit wie bei dem Unterprogramm „ad\_in(x)“ kommt.

Timer-Y ruft auch das Unterprogramm „gkevents()“ auf, um so den Giessstart, die Giesskurvendaten und das Giessende samt Giesszeit und Giestakt ermitteln zu können. Parallel dazu werden u.a. auch die maximale und mittlere Giesshöhe ermittelt.

Timer-X wird erforderlichenfalls über einen Steuercode gestartet und ist ausschließlich für die Steppermotorsteuerung vorgesehen, die zu einem späteren Zeitpunkt zur Anwendung kommt. Die Frequenz dieses Timers ist variabel vorgesehen und wird zeit- und giesskurvenabhängig berechnet und gesetzt.

Innerhalb „gkevents()“ wird „txmode“ gesetzt. Je nachdem die aktuelle Giesssituation ist, wird „txmode“ gesetzt, was wiederum die zu übertragenden Daten steuert. Das Giessen wird also in „ruhend“, „Giessstart“, „aktives Giessen“ und „Giessende“ unterteilt.

Für die LCD-Anzeige (240\*128) ist zwar schon Code für das grafische Setzen von Zahlen, Zeichen und Linien implementiert aber noch nicht abschließend getestet.

### **Bemerkung:**

Es wurde versucht, sehr viel Information im Listing selbst unterzubringen, so dass das Listing größtenteils selbsterklärend ist. Natürlich ist auch der Code optimierungsfähig. Viele Auskommentierungen sind teilweise nützlich oder / und überflüssig.

### **Bildnotizen:**

In Bild 1 ist schematisch die Giessschnauze mit einem zu füllenden Formkasten wiedergegeben. Im linken Bildteil ist die Stopfenstange erkennbar, welche zur Formkastenbefüllung mit einem Hydraulikzylinder angehoben wird und somit den Ausfluss freigibt. Im rechten Bildteil ist der Schwimmer erkennbar, dessen Position mit Näherungsschalter abgetastet wird. Die Formfüllung verursacht eine Absenkung des Giessniveaus in der Schnauze. Der sich leerende druckbeaufschlagte Ofen wird automatisch durch Druckerhöhung im Kessel auf ein möglichst konstante Schnauzenhöhe eingestellt.

Die provisorische Stopfenhubmessung mittels eines Drehpotis in Spannungsteilermanier zeigt Bild 2. Mit einem Magnet ist das Provisorium einerseits am waagrechten Teil befestigt und abgewinkelt hierzu ist der Mitnehmer durch einen Magnet mit dem Hubzylinder verbunden.

Auf die gleiche Art wird ein Drehpoti zur Schwimmerkontrolle eingesetzt, wie dies Bild 3 zeigt.

Bild 4 zeigt die mit einem Magneten auf der Giessschnauze plazierte Webcam, welche die Giesstrichterbilder schiesst, wie sie die Bilder 5 und 7 wiedergeben.

In Bild 6 ist die Messplatine auf einer Lochstreifenplatine dargestellt. Der Aufbau entspricht dem Schaltplan aus dem Dezemberheft ergänzt um die Bauteile für eine Spannungsstabilisierung mittels einem 7805. Die verfügbaren Portanschlüsse sind nahe am IC-Sockel liegend auf Steckstifte geführt. Zusätzlich wurden ausreichend Abnahmestifte für Spannungsabnahmen vorgesehen.

Das Ergebnis ist die in Bild 7 gezeigte Grafik. Hier werden im linken oberen Bildteil die Giesskurve selbst und darüberliegend das Verhalten des Schwimmers während des Giessvorganges dargestellt. Je nach Wunsch können mehrere Giesskurven aufeinanderliegend farblich unterscheidbar angezeigt werden. Im linken unteren Bildteil ist der Giesstrichter des letzten Abgusses eingeblendet. Das Rechteck um den Trichter entspricht einer Fokussierung desselben. Dies ist der Bereich, der im rechten Bildteil unter optische Höhe als Streifenbild pro Kasten eingezeichnet wird. Unter statische Giesshöhe wird die per Pixelauswertung ermittelte Giesshöhe im Formkasten eingetragen. Im rechten Bildteil werden von oben nach unten zunächst die Giesszeit, kombiniert die maximal und mittleren Stopfenhübe, das Schwimmverhalten und das rechnerische Giessgewicht grafisch kastenbezogen dargestellt. Zusätzlich wird rechtsseitig passend die effektive Kastenzahl, die letzte Giesszeit, Giesstakt, mittlere und max. Hubhöhe u.a. in Textform angezeigt.

Bild 8 zeigt die Steckverbindung, die es erlaubt, das LCD-Display parallel an einen PC anzuschliessen oder aber an den 8-Bit-Port 1 der MPU. Als Steuerleitungen benötigt das 240x128 Pixel grosse Display zusätzlich 4 Steuerleitungen.

In Bild 9 ist die provisorische Verbindung zwischen MPU und Display zu sehen.

In Bild 10 wird versucht, beispielhaft eine Giesskurvenmessung auf dem LCD-Modul anzuzeigen. Die Anzeige ist mit Visual-Basic erzeugt und noch nicht vollständig und fehlerfrei im Listung der MPU wiedergegeben.