Fachdidaktik - Lernaufgabe

„Einführung in das Studium der Berufspädagogik“

Die Programmierung eines Blinkcodes auf Basis des Arduino Uno

Eingereicht: Pädagogische Hochschule Tirol

Betreuer: Ing. Bed. M.A. Lagger Martin

Eingereicht von: Mag. (FH) Konrad Andreas

Innsbruck, den 03.11.2013



**Eidesstattliche Erklärung**

Ich erkläre an Eides statt, dass ich die vorgelegte Arbeit zur Erfüllung der   
Modulanforderung selbst verfasst und keine anderen als die angeführten Behelfe verwendet habe. Ich bin damit einverstanden, dass die Arbeit unter Wahrung aller Urheberrechte anderen Lehrer/innen zugänglich gemacht wird.

Innsbruck, den 3.11.2013

Mag. (FH) Andreas Konrad

|  |  |
| --- | --- |
| **Thema** | Blinkcode |
| **Kompetenzen** | **Fachkompetenz**   * Der Schüler/-in arbeitet selbständig mit Online Hilfen. * Der Schüler/-in kann den Syntax lesen und die Musterbeispielprogramme selbständig umsetzen. * Der Schüler/-in versteht das Konzept der Beschaltung von Digitalen Ein- und Ausgängen des Mikrocontrollers, bzw. des Mikrocontrollerboards * Der Schüler/-in baut in Kleingruppen eine Schaltung auf und programmiert diese |
| **Methodenkompetenz**   * Der Schüler/-in kann Hilfetexte sinnerfassend lesen und deren Inhalte anwenden * Der Schüler/-in kann Aufgaben selbständig und in Kleingruppen * Der Schüler/-in kann Inhalte praxisgerecht anwende |
| **Sozial- und Personalkompetenz**  Der Schüler/-in arbeitet in Kleingruppen und erfasst Problem- und Hilfestellungen selbständig.  Der Schüler/-in diskutiert mit seinen Kollegen/-innen Vorgehensweisen und Lösungsansätze. |

|  |  |
| --- | --- |
| **Lehrplanbereiche** | Mechatronische Technologie, Elektroniklabor  **Kernlehrstoff:**   * Mikrocontroller   + Hardwarebeschaltung   + Softwareprogrammierung   + Digital I/O Beschaltung   + Programmprobleme durch NOP Loops * LED   + Richtige Bauteilplatzierung (A/K)   + Berechnung des Vorwiderstandes * Timerprogrammierung (angerissen)   + Timer Grundlagen   + Anwendung   + Interrupt   + Interrupt Service Routinen   **Lehrplanbezug:**  Mikroelektroniklabor, 8. Klasse |
| **Zielbereiche** | Der Schüler/-in lernt technische Zusammenhänge  Der Schüler/-in verknüpft Hard- mit Software  Der Schüler/-in agiert als integraler Bestandteil mit den Gruppenmitgliedern  Der Schüler/-in generiert durch selbständiges Arbeiten am Problem unter Anwendung von Hilfen, durch Lehrer und Texte, Wissen im interdisziplinären Kontext. |

|  |  |
| --- | --- |
| **Didaktisch - methodische Hinweise** | Der Schüler/-in müssen Vorwissen im Rahmen von Mikrocontrollertechnik mitbringen  Dieses sollte folgendes umfassen:   * Vertrautheit mit der Arduino Entwicklungsumgebung * Vertrautheit mit rudimentären Elementen der Programmiersprache C * Berechnungsverfahren zur Ermittlung von Vorwiderständen * Beschaltung von digitalen Ein- und Ausgängen * Anschluss von Schaltern an ein Mikrocontrollerboard * Die Vertrautheit mit dem Oszilloskop   Zur Ausarbeitung der Problemstellungen soll die Online Reference von Arduino herangezogen werden. Der Lehrer entscheidet, ob er Fragen beantwortet oder auf die Online Hilfe oder den empirischen Ansatz verweist.  Den Arduino Einsteigersets liegen Einführungsbücher bei. Der Lehrer entscheidet, ob diese Bücher herangezogen werden.  Die Aufgaben werden in Kleingruppen ausgearbeitet und gelöst.  Die Kontrollfragen dienen der Überprüfung, ob der Schüler oder die Schülerin die jeweiligen Aufgaben gelöst hat.  Für den Lehrer liegen die Lösungen der Aufgaben diesem Papier bei. |

|  |  |
| --- | --- |
| **Schwierigkeitsgrade** | Leicht C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0252349.wmf  Mittel C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0252349.wmfC:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0252349.wmf  Schwer C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0252349.wmfC:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0252349.wmfC:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0252349.wmf |
| **Zusatzmaterial und Unterlagen** | **Für Schülerinnen**  Arduino Online Referenz auf Englisch und Deutsch  <http://arduino.cc/en/Reference/HomePage>  <http://arduino.cc/de/Reference/HomePage>  Let’s make robots: (Timerprogrammierung)  [www.letsmakerobots.com/node/28278](http://www.letsmakerobots.com/node/28278)  Buch: Arduino für Einsteiger (O’Reilly, 2012)  ISBN: 978-3-86899-232-8  **Für Lehrer und Schülerinnen über den Unterricht hinaus**  Buch: AVR – RISC Mikrocontroller (Franzis, 2000)  ISBN: 3-7723-5475-0  Buch: 30 Adruino Selbstbau-Projekte (Franzis 2012)  ISBN: 978-3-645-65136-3  In dieser Lernaufgabe befinden sich **keine** Bilder, die vor der Verwendung im Unterricht urheberrechtlich geprüft werden müssen! |

|  |  |
| --- | --- |
| **Zeitbedarf** | **Drei Unterrichtseinheiten zu je 50 Minuten (150‘)**  Einleitung 10‘  Grundgerüst 25‘  Programmierung – Wiedereinstieg 15‘  Signalanalyse, Programmierung, Aufbau 75‘  Programmierung, Kontrolle, Überlegungen 25‘  **Ges. 150‘**  Je nach Vorbildungsgrad kann die Zeitliche Einteilung stark schwanken. Die Bandbreite könnte von 50‘ bei fortschrittlichen Klassen, bis 150‘ bei Neulingen reichen. |

**Blinkcode, realisiert durch die Verwendung eines Mikrocontrollers**

In der Digitaltechnik werden Informationen oft seriell als Code übertragen. Fernsehfernbedienungen auf Infrarotbasis übertragen zum Fernseher verschiedene Pulsmuster, welche vom dortigen Empfänger interpretiert werden.

Das Signal für Laustärke erhöhen unterscheidet sich z.B. vom Signal des Kanalwechsels durch sich wiederholende Pulsmuster.

Das folgende Steuersignal wurde durch einen Singelshot eines Digital Speicher- Oszilloskops aufgenommen.

Der längste Low Pegel zwischen den Signalpaketen, die sich periodisch wiederholen, ist die Pausenzeit zwischen den Signalpaketen. Die Aufzeichnung beginnt beim senkrechten Pfeil. (Trigger Start)



**Lernsetting 1:**

**Reproduziere das obige Signal an einer LED mit Hilfe passender   
Elektronik und eines Oszilloskops.**

|  |  |
| --- | --- |
| **Grundgerüst** | **Schritt 1: 5‘**  Überlege in Partnerarbeit auf Basis von früheren Unterrichtsstunden, welche elektronischen Komponenten für das codierte Blinklicht verwendet werden können.  C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0252349.wmfDie Hardware entspricht einer einfachen Blinkschaltung.  Die Codierung erfolgt durch Programmierung des µC’s.  Arduino Uno Board  Vorwiderstand und LED |
| **Schritt 2: 5‘**  C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0252349.wmfÜberlege in Partnerarbeit, worauf beim Betrieb einer LED zu achten ist?  Vorwärtsspannung und nötige Stromstärke |
| **Schritt 3: 15‘**  C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0252349.wmfÜberlege in Partnerarbeit, wie der Vorwiderstand zum Betrieb der LED ausgelegt werden kann. (Skizze und Berechnung bitte hier) |
| **Programmierung** | **Schritt 4: 15‘**  C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0252349.wmfSchließe in Einzelarbeit das Arduino Uno Board an den Computer an und starte die Entwicklungsumgebung. Wenn Probleme auftreten, wende dich zuerst an KlassenkollegInnen.  Ev. wird das Board nicht richtig erkannt oder es wird die falsche COM Schnittstelle zugewiesen. Wenn das der Fall ist, muss über den Gerätemanager eingegriffen werden.  **Wiederholung:**  Ganz oben werden den Board Pins Variablennamen zugeordnet.  Im Programmteil void setup (){} wird der Code geschrieben, der nur ein einziges Mal beim eigentlichen Programmstart ausgeführt werden soll. Z.B. die Zuordnung der Pin Variablen zu Aus- oder Eingängen.  Im Programmteil void loop () {} wird der Programmcode geschrieben, der zyklisch wiederholt werden soll.  Hilfe zum Wiedereinstieg unter:  <http://arduino.cc/en/Reference/DigitalWrite> |

|  |  |
| --- | --- |
| **Signalanalyse, Programmierung und Hardwareaufbau** | **Schritt 5: 15‘**  C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0252349.wmfC:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0252349.wmfAnalysiere in Partnerarbeit den zeitlichen Verlauf des oben abgebildeten Signals.  50ms ein, 25ms aus, 100ms ein, 150ms, aus  **Schritt 6: 10‘**  C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0252349.wmfEntwerfe zur besseren Übersichtlichkeit in Einzelarbeit eine Tabelle, die die zeitliche Abfolge des Signalverlaufes aufzeigt.  **Schritt 7: 25‘**  C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0252349.wmfVersuche in Einzelarbeit das analysierte Signal programmtechnisch zu realisieren. .  Das Signal soll mit gleicher Pausenlänge zwischen den Signalblöcken periodisch erzeugt werden. Ein Start Taster ist nicht vorgesehen.  Programm im Anhang.  **Schritt 8: 15‘**  Übertrage in Einzelarbeit das Programm auf das Arduino Board und baue auf dem Breed Board eine Schaltung mit einer LED auf.  C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0252349.wmf  **Schritt 9: 10‘**  C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0252349.wmfSchließe zu zweit das Oszilloskop an den passenden Punkten an und überprüfe das programmierte Ergebnis auf Übereinstimmung mit der Signalvorgabe.  C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0252349.wmf |
| **Programmierung, Kontrolle, weitere Überlegungen** | **Schritt 10: 10‘**  Überlege in Einzelarbeit, wie das Programm mit der Delay() Funktion arbeitet.  C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0252349.wmfC:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0252349.wmfWelches Problem würde sich ergeben, wenn zusätzlich zur Code LED eine LED zur Funktionskontrolle periodisch mit einer Frequenz von 1,908Hz blinken soll?  Das Problem mit der Delay Funktion liegt darin begründet, dass ein Delay das ganze Programm für die angegebene Zeit stoppt. Die zyklische Wiederholung des Hauptprogrammes wird dadurch unterbrochen, es können während der Unterbrechung keine anderen Befehle ausgeführt werden.  Müsste eine weitere LED periodisch blinken, würde sich das mit Delay Funktionen nicht realisieren lassen.  Abhilfe schafft hier die Programmierung durch frei laufende und Interrupt gesteuerte Timer. Dasselbe Problem würde sich bei einer S7 SPS ergeben.  **Schritt 11: 15‘**  Diskutiere deine Erkenntnisse mit Kollegen und Kolleginnen und stelle einen Zusammenhang mit hypothetischen Delay Befehlen bei einer S7 her.  C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0252349.wmfWählt ein Zweierteam, das diese Erkenntnisse kurz vor der ges. Klasse präsentiert.  C:\Program Files\Microsoft Office\MEDIA\CAGCAT10\j0252349.wmf  Aus diesem Grund werden im Bereich der S7 Zeitbeeinflussungen nicht durch delay() Funktionen beschrieben, sondern durch Timerprogrammierung. |

**Programm zu Aufgabe 7**

int LED = 7;

void setup()

{

pinMode(LED, OUTPUT);

}

void loop()

{

digitalWrite(LED, 1);

delay(50);

digitalWrite(LED, 0);

delay(25);

digitalWrite(LED, 1);

delay(100);

digitalWrite(LED, 0);

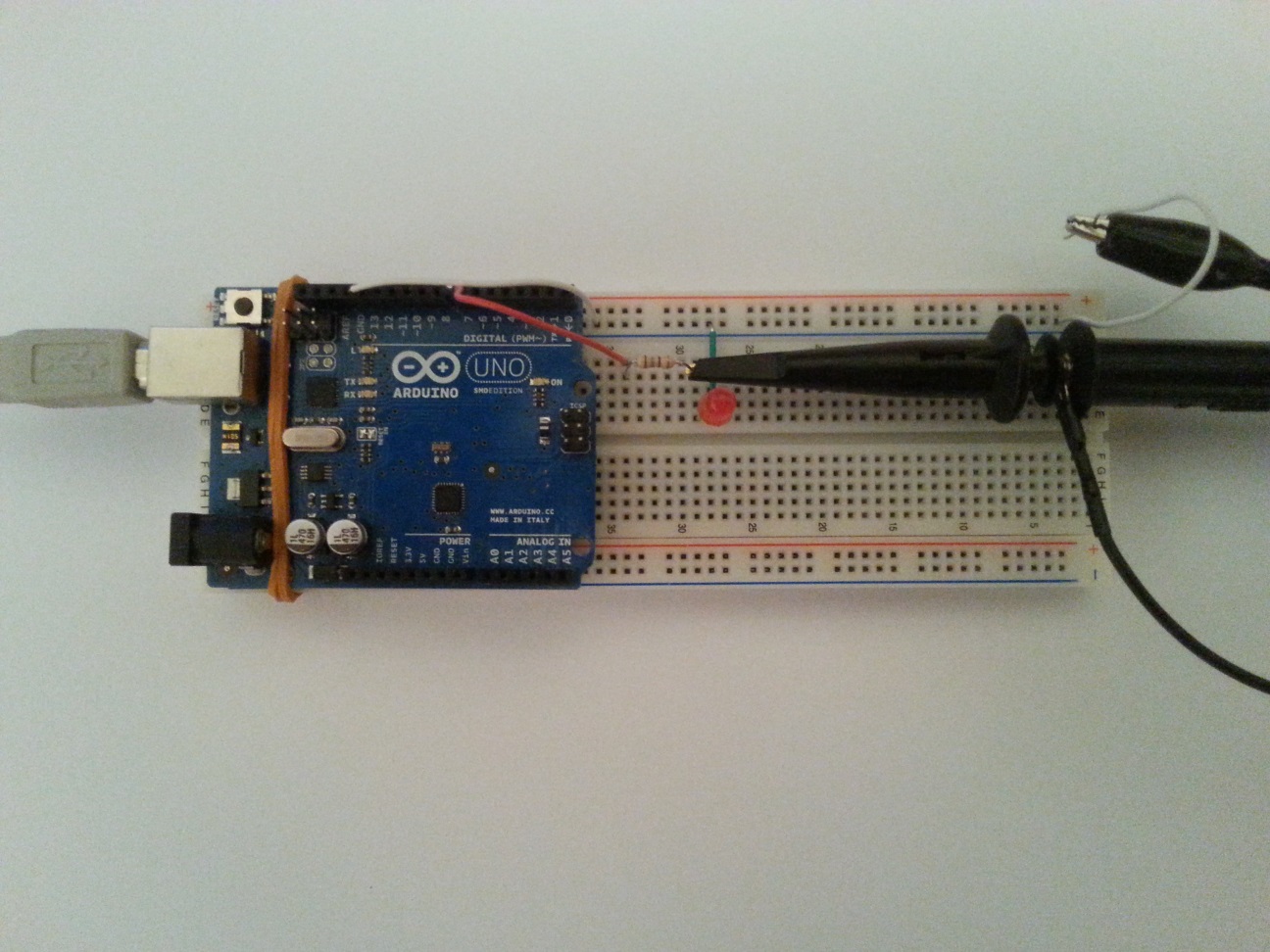
delay(150);

}

**Signaltabelle zu Aufgabe 6**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LED | ON | OFF | ON | OFF |
|  |  |  |  |  |
| Zeit in ms | 50 | 25 | 100 | 150 |
| Zeit ges. in ms | 50 | 75 | 175 | 325 |

**Aufbau zu Aufgabe 8 und 9**

****