

Embedded Systems I

Lab 4: Hausautomation Teil 1 - Intelligente Lampensteuerung

Beschreibung:

In dieser Übung werden Sie den Arduino zur Kontrolle von einer Lampenschaltung in einem Haus (simuliert durch mehrere LEDs) verwenden. Die Lampen sollen in einem bestimmten zeitlichen Verlauf an- und ausgeschaltet werden und verschiedenen Helligkeitsstufen erlauben.

Aufgaben:

1. Verschalten Sie 4 LEDs an die Anschlusspins 4-7 des Arduino Boards. Schreiben Sie ein Programm, das die LEDs gleichzeitig für einen festen Zeitraum ein- und dann wieder ausschaltet. Simulieren Sie dabei einen Tagesverlauf, wobei eine Stunde einer Sekunde entspricht. Die Lampen sollen wie folgt an- und ausgeschaltet werden:
 - LED 1 (Lampe Keller): An von 6:00-22:00Uhr, sonst aus.
 - LED 2 (Lampe Wohnbereich): An von 6:00-8:00Uhr und von 18:00-22:00Uhr, sonst aus.
 - LED 3 (Lampe Flur): An von 6:00-9:00Uhr und von 17:00-23:00Uhr, sonst aus.
 - LED 4 (Lampe Küche): An von 7:00-20:00Uhr

Verwenden Sie für jede LED einen Vorwiderstand (220Ω oder 330Ω) um die LED und das Arduino Board nicht zu zerstören.

Nutzen Sie die Funktion „delay()“ mit einem Übergabeparameter in Millisekunden, um das Arduino Board zu pausieren.

Wichtig: Das Arduino System pausiert bei Erreichen der „delay()“ Funktion vollständig. Es ist also nicht möglich, parallel Befehle auszuführen (zum Beispiel Taster Zustände über „digitalRead()“ einlesen).

2. Erweitern Sie das Projekt um eine Ausgabe über die serielle Schnittstelle. Über diese Ausgabe soll zu Beginn eine Begrüßung ausgegeben werden ("Hausautomation Version 1.5" o.ä.) und im Verlauf des Programmes sowohl die simulierte Uhrzeit in Stunden sowie der Zustand der Lampen.

Hinweise:

- Um eine serielle Verbindung mit dem Computer herzustellen, muss die Verbindung innerhalb der „setup()“ Funktion initialisiert werden. Die Funktion „Serial.begin()“ erwartet als Übergabeparameter die Geschwindigkeit der Datenverbindung als Integerwert
Beispielaufruf: `Serial.begin(9600);` *//initialisiert eine serielle Verbindung mit 9600 baud (bit/s)*
 - Die Funktion „Serial.println()“ sendet den an sie übergebenen Datenstring (*val*) über die serielle Schnittstelle zu einem Empfangsgerät.
Beispielaufruf: `Serial.println("TEST 1!");` *//schreibt auf Datenleitung den Datenstring TEST 1!*
 - Um die gesendeten Daten der seriellen Schnittstelle anzeigen zu können, verfügt die Arduino Entwicklungsumgebung über ein eingebettetes Tool. Klicken Sie zum Öffnen auf den Reiter „Tools“ → „Serial Monitor“. Um die Lesbarkeit der Ausgabe auf dem Bildschirm zu erhöhen, können Sie eine Wartezeit über die Funktion „delay()“ implementieren.
3. Zusatzaufgabe: Erweitern Sie das Programm um einen Wochenmodus, bei welchem die Wochentage Mo-Fr wie in Aufgabe 1 geschaltet sind und das Wochenende Sa-So anders geschaltet wird.
 4. Zusatzaufgabe: Erstellen Sie ein neues Projekt, in welchem Sie eine sich dimmende LED mit einer Periodendauer von 2s realisieren. Die LED soll dabei innerhalb der Periodendauer ihren Zustand von hell auf dunkel und umgekehrt langsam ändern, so dass ein Effekt ähnlich dem einer sich dimmenden Lampe entsteht