Lab assignment 3: Digital Input

When we talk about switches, we often think of a physical switch that has two states, on and off. A button is also a switch, but if we see it as a pressure sensitive sensor, we can see many different applications. For example, why turn the lights on and off with a light switch on the wall, why not consider the switch as a pressure sensitive sensor and turn the lights on and off by registering if somebody is sitting on the pressure sensitive sensor build in the sofa. And when we replace the pressure sensitive sensor by a different actuator we can make many more new applications. e.g. movement sensor, light sensor, sound...

This lab assignment we use two different sensor. We start with a pressure sensor (button) and end with a movement sensor (PIR)

Button​(pressure​sensor)

In the lecture is discussed that we can connect buttons to the Arduino. The Arduino.cc website also describes how to properly connect a button. Follow these steps. http://arduino.cc/en/Tutorial/Button

Input

Currently all our code behavior is determined “at compile time”. After compiling the code the behavior of the Arduino won’t change anymore. In this exercise we are going to expand our code behavior to make a more dynamic program.

Expand your code to register how many times you have pressed the button. Further create a drawNumber() function and use this function to display the times pressed in the terminal view every time when the button is pressed.

You may notice that sometimes the counter seems to count multiple button presses when you press the button only once. This is an undesired behaviour.

How do we call this behavior?

Bounce

Describe two ways how you can remove this behavior with their advantages and disadvantages

1. Use a buffer, in 3 possible ways: 1) use an array 2) use 2 variables, the first counts the amount of state changes, the second stores the value 3) use an algorithm Pros: Most accurate Cons: most performance heavy
2. Use timing, so a state is being active for atleast a certain amount of time, Pros: Easy implementation Cons: not really accurate/responsive

.........................................................................................................................

..........................................................................................................................

..........................................................................................................................

............................................................................................................................

..........................................................................................................................

..........................................................................................................................

.......................................................................................................................

Chose one debounce option and implement it in your Arduino code.

Still experiencing problems? Use the following debug list.

Problem help

The counter isn’t working First check your connected hardware,

Then check your code step by step, maybe you missed a step, or aren’t looking at the correct input pin. Next check if some hardware is broken

The counter is running even when I don’t press the button

You have a floating pin

The counter increases to much when I keep the button pressed

You are not registering press continous, only register change of press

The counter is increasing with multiple after one press

still experiencing button bounce

Input​display

We expand our exercise by using the LED matrix. Download the Counter.ino from the VLO

● Run the sketch

Adapt the sketch to display the counted button presses instead of seconds. Hint you can use the drawNumber() function.

Reflex

Let’s make a multiplayer reflex game in 2 steps

Step 1. Adapt your code to make the following possible: - connect two buttons to your Arduino - the code should register which button is pressed and which button is released - when a button is released, the LED matrix should display this button via a number or

other identification marker You can now make a simple reflex game with two players and one referee. When the referee makes a sound the two players should release their pressed button as fast as possible. The last to release is displayed on the screen and the slowest.

Step 2. In this step we will replace the referee by code. Expand your code to make the full game, implement the following functionality:

- the game should be started by usage of the buttons and feedback about the game

starting should be displayed on the LED matrix - The LED matrix should display when the players should release the buttons - For the release signal, a random waiting time should be used between 1 and 10

seconds - The LED matrix should display the winner (the player who released the button closest

to the release signal) - When a player released his button before the signal, a red cross should be displayed.

Bewegingssensor​(PIR-sensor)

The reflex game is fun using buttons, but we can easily adapt it using different sensors. This exercise focuses on using a PIR sensor instead of buttons.

A​PIR?

Find out what a PIR sensor is by reading the Wikipedia article until the ‘Mirors’ section http://en.wikipedia.org/wiki/Passive\_infrared\_sensor

Think of you a PIR sensor can be used as a movement sensor. Describe the workings of the PIR sensor als movement sensor by starting with a space without movement until displaying the registered movement in the terminal window of the Arduino.

Use the words:

infrared, signal, input, high, Arduino, terminal, movement, difference

Step 0 No movement, PIR sensor gives low input signal to the Arduino

Step 1 ........................................................................................................

.......................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

...........................................................................................................................

Try to implement your steps in a new Arduino sketch, and check if you missed any steps, or could describe some steps in more detail.

If you didn’t succeeded in describing all steps you can find a PIR example on the Arduino.cc website. Try to understand the code before you start implementing it in you own sketch.

Reflex​New-And-Improved!

Now adapt your reflex game to react on movement of the PIR sensors instead of buttons. You can use your hand or boxes to cover the sensor and distinguish between movement and no movement. When the LED matrix displays the signal to release you release your hand (or lift the box) of the PIR sensor. Tip: Our PIR sensors calibrate when started. So let it calibrate when you have your hands (or box) covering the sensor.

Time left for an extra exercise?

● Implement a score in your game. Display the scores of both players as dots in the corners. Players gain a dot by winning a game and lose a dot by moving to early (except when they own no dots)