Embedded pool

Rush 00: Multiplayer

contact@42chips.fr

Summary: *Introduction to integrated circuit communication*

Version: 1

Chapter I

Introduction

`Tennis for Two` is a tennis video game designed in 1958 by William Higinbotham and developed by Robert Dvorak, often considered as the first video game in history.
The game was developed on a dedicated Donner Model 30 analog computer connected to an oscilloscope, which serves as the screen. The concept was presented in 1958 and 1959 during the open house of the Brookhaven National Laboratory.

The game, developed in just a few weeks, was designed to showcase technological advancements to the general public, during the laboratory's open house event.
During the Cold War, Higinbotham wanted to show a more friendly image of the world of research, as minds were still marked by the first nuclear tests of the 1950s.
Using documents on ball trajectory simulations, he created an entertainment concept without any particular academic utility, using an oscilloscope and an analog computer.

The game received a good reception from the local public, where queues formed to access the machine, and it was presented to the public for two consecutive years (source).
Despite the ingenuity of the concept for its time, the game was not used after 1959, as the laboratory decided to replace this attraction with another one during the next open house event and to reuse the parts for other research.

> These exercises are to be done in pairs.

> Both of you must understand how the whole code works.

Chapter II

General instructions

Unless explicitly stated otherwise, the following instructions will be valid for all assignments.

- The language used for this project is C.

- It is not necessary to code according to the 42 norm.

- The exercises are ordered very precisely from the simplest to the most complex. Under no circumstances will we consider or evaluate a complex exercise if a simpler one is not perfectly successful.

- You <u>must not</u> leave <u>any</u> files other than those explicitly specified by the exercise instructions in your directory during peer evaluation.

- All technical answers to your questions can be found in the `datasheets` or on the Internet. It is up to you to use and abuse these resources to understand how to complete your exercise.

- You <u>must</u> use the datasheet of the microcontroller provided to you and comment on the important parts of your program by indicating where you found the clues in the document, and if necessary, explaining your approach. Don't write long blocks of text, keep it clear.

- Do you have a question? Ask your neighbor to the right or left. You can ask in the dedicated channel on the Piscine's Discord, or as a last resort, ask a staff member.

Chapter III

Mandatory part

	Exercise 00
	Howdy cowboy
Turn-in directory : *ex00/*	
Files to turn in : `Makefile, *.c, *.h`	
Allowed functions : `avr/io.h, util/delay.h, util/twi.h`	

III.1 Objective

The objective of this exercise is to create a speed game where 2 microcontrollers communicate with each other. You should use 2 of your boards and connect them with the provided cable. The point of this exercise is to show you how communication with 2 MCU can work.

> Do **NOT** try to push something in a place where it doesn't want to go.
> Otherwise you will damage your board.

III.2   Requirements

> You don't have to implement the I2C protocol for this rush.
> You only need to fit the requirements of the exercise.

- Connect the 2 microcontrollers using the provided cable on the I2C interface of both boards.

- The AVR ATmega328P microcontroller has 1 I2C peripheral that you must use in this exercise to communicate with another microcontroller.

- The MCU's I2C must be configured with a frequency of 100kHz.

- You must decide which microcontroller is the master and which is the slave.

- Both MCUs must have the exact same code flashed.

- The MCUs must be able to communicate if one of them restarts. You may decide to restart the game.

III.3   Game rules

The game is rather simple:

- When both players press the button, the game starts.

- A countdown is displayed on the LEDs.

- The fastest player to press the button at the end of the countdown wins the game.

- Another game can now start.

> If a player presses the button before the end of the countdown,
> they lose the game. You should create a light signal indicating the
> winner and the loser. Don't forget to comment the rules for grading!