Fachhochschule
Münster University of
Applied Sciences

Lab for Semiconductor Devices and Bussystems

Exercise Embedded Systems WS2012/13
Tutor: Dipl. Informatiker Gregor Rebel

Exercise 4
Application

The goal of this exercise is to extend the graphical application built in Exercise 3.
For this, every group gets an Olimex STM32-LCD prototype board.

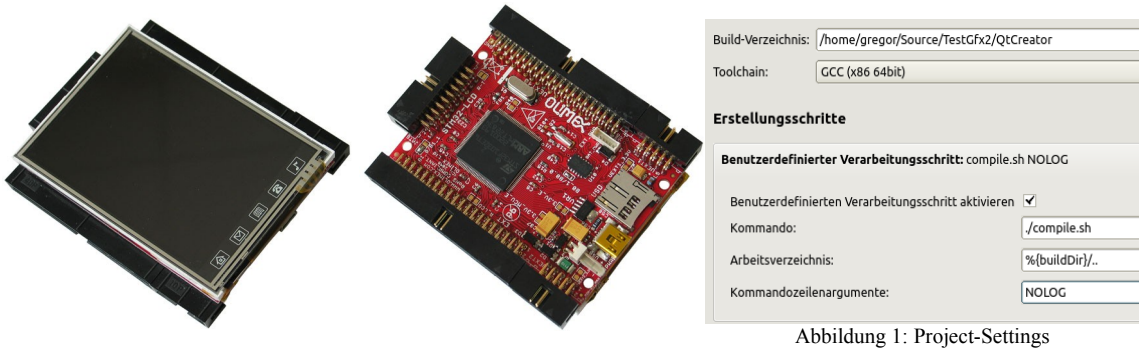

Abbildung 1: Project-Settings

You start by creating a new project with newest version of The ToolChain and enabling these lines in your activate_project.sh:

- activate.100_board_olimex_lcd.sh
- activate.600_example_gfx_boxes.sh

Don't forget to deactivate other boards and examples. After a first compile run, you should see lots of moving boxes on the screen.

After loading the project into QtCreator, make sure that your project settings are correct for compilation:
Now open ../examples/example_gfx_boxes.h and switch to the source code (F4). For this exercise you may directly code type your changes into this file.

1. In the previous Exercise, the graphical boxes were reflected at the screen borders. The boxes now shall be reflected by each other too. For this, a global datastructure shall track all rectangle positions and sizes. You may reduce the amount of boxes to 10 for a smoother display. Keep in mind, that a 32-bit architecture can load/ store 32 bit values atomically.

2. Make collisions more realistic by giving each box a weight that is calculated as the amount of pixels occupied by it. If two boxes collide, each brings in an impulse that is proportional to its weight and speed.

3. The STM32-LCD board has an onboard 3D accelerometer being connected to the CPU via an I2C-bus. Create another Task tAccelerometer() that periodically reads in the acceleration data in X- and Y-axis and writes them into global variables s32_t AccelerationX, AccelerationY. Each tBox() task should now take these values into account to change the current SpeedX and SpeedY. Make it look and feel like if the boxes were real blocks in a glass box.
   You can find example code for accessing the Accelerometer in ../example/example_i2c.c. Simply enable activate.600_example_i2c_master.sh to see how it works.

One important function for this exercise is ttc_gfx_rect_fill() as described in ../common/ttc_gfx.h.

Below you can see an overview of the dataflow and to be implemented. The tBox() and tDraw() tasks are already implemented in the example code. Modify the to suit your needs.Figure 3 shows example vectors of moving and colliding boxes. The yellow and blue box in the middle show different secondary (light colored) vectors according to their different masses.

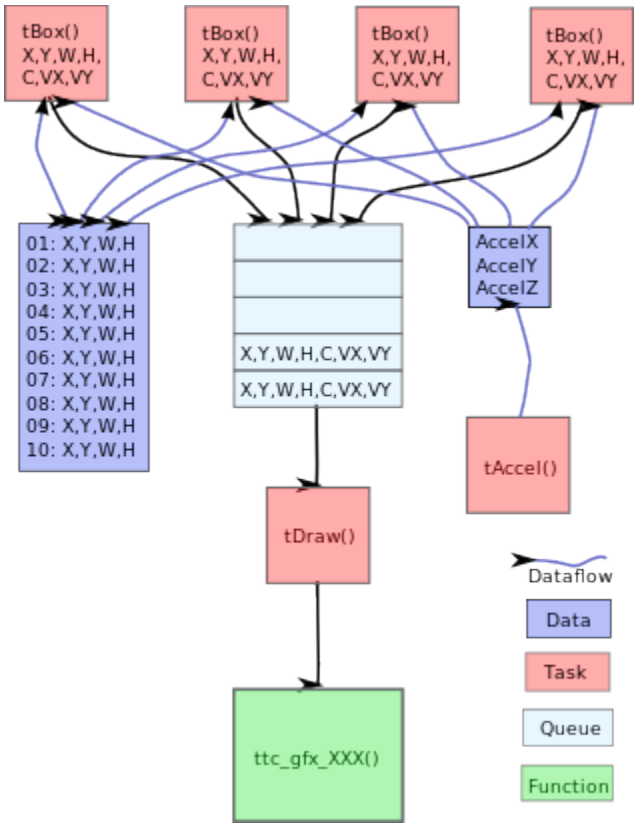

Figure 2: Dataflow between Tasks

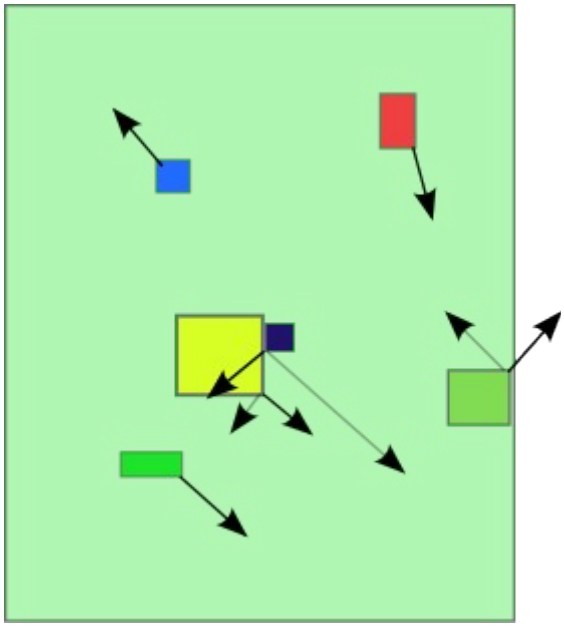
Figure 3: Boxes colliding with each other