

Embedded Systems Engineering

Project B: Pacman Firmware Overview

R00159222 - Zachary Dair
zachary.dair@mycit.ie

This document provides a brief overview of the firmware used in the Project B - Pacman assignment.

The **mk** file acts as the launcher for the application, when executed in the terminal it results in the opening of QEMU emulating our application's display.

Inside the **mk** file, the **.o** files are created from our **.bmp** images, resources are later used to display walls, sprites, points etc.

After these **.o** files have been created a toolchain is executed which compiles and creates the required binaries, after completion the script waits for input before finally connecting to QEMU and emulating the visual display that would be present on a real board.

The next significant part of the firmware is the actual application code itself, this is mainly split into three sections, **t.c**, **vid.c** and **dijkstra.c**.

The **t.c** file contains the core application code, such as the initial map data, all initial setup such as initialising the positions of both pacman, and the enemy sprites, drawing the map, constructing the relationships between vectors on the map, a main while loop to continuously run the pacman code unless the game has been won, or lost.

t.c also employs the **dijkstra.c** code for enemy pathing, **vid.c** for display, **uart.c** to handle user input and certain aspects of display, **timer.c** for tick counting, and **interrupts.c** to read status registers and discern which to be interrupted, the UART0 and UART1 RXIM interrupts are enabled, as are the timer interrupts.

vid.c then handles the pixel mapping side of the display, it can draw sprites from the compiled external char object, it is also used to replace the overwritten pixels as the sprites move