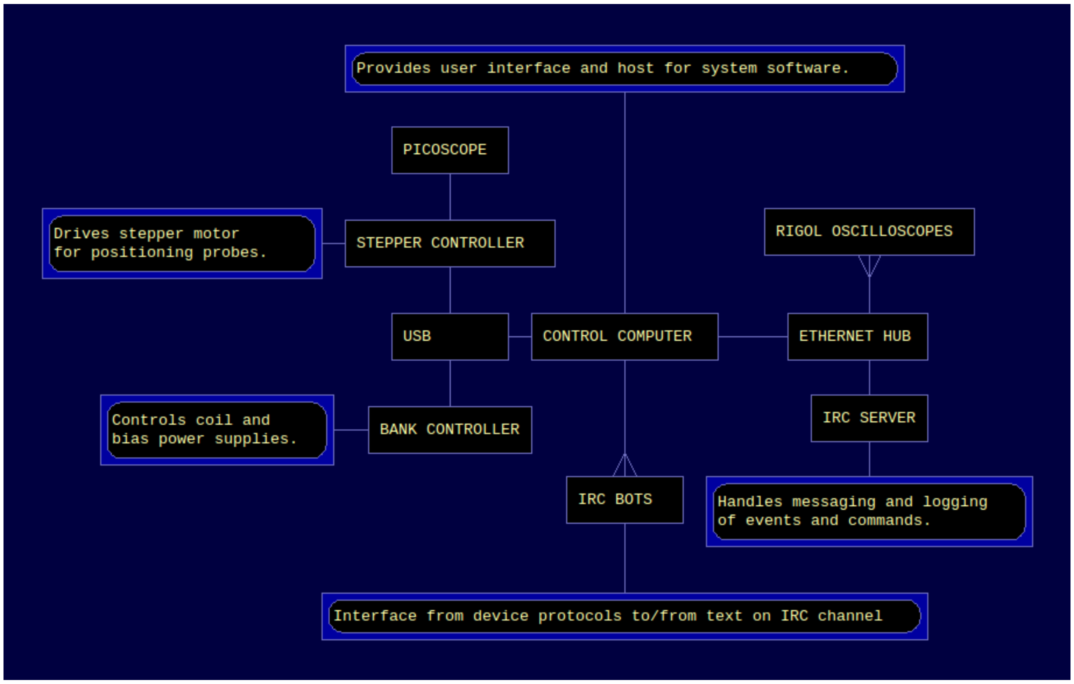

Figure 1: Overall system design

# 1 The Control System

This section describes the embedded control system and user interaction. The latest system software and documentation can always be found on github at `https://github.com/hedj/fusion/`.

Flowcharts in this section use the visual language from the DRAKON project. See `http://drakon-editor.sourceforge.net/language.html` for details.

## 1.1 The Bank Controller

The bank controller is a Controllino Mega; an Arduino-based programmable-logic controller. The bank controller is in charge of the following functions:

- Capacitor-bank and HV pulse generation
- HV voltage control
- Capacitor bank charge control
- Audible alert before pulse

The top-level flow of the bank controller is shown in Fig. 2.

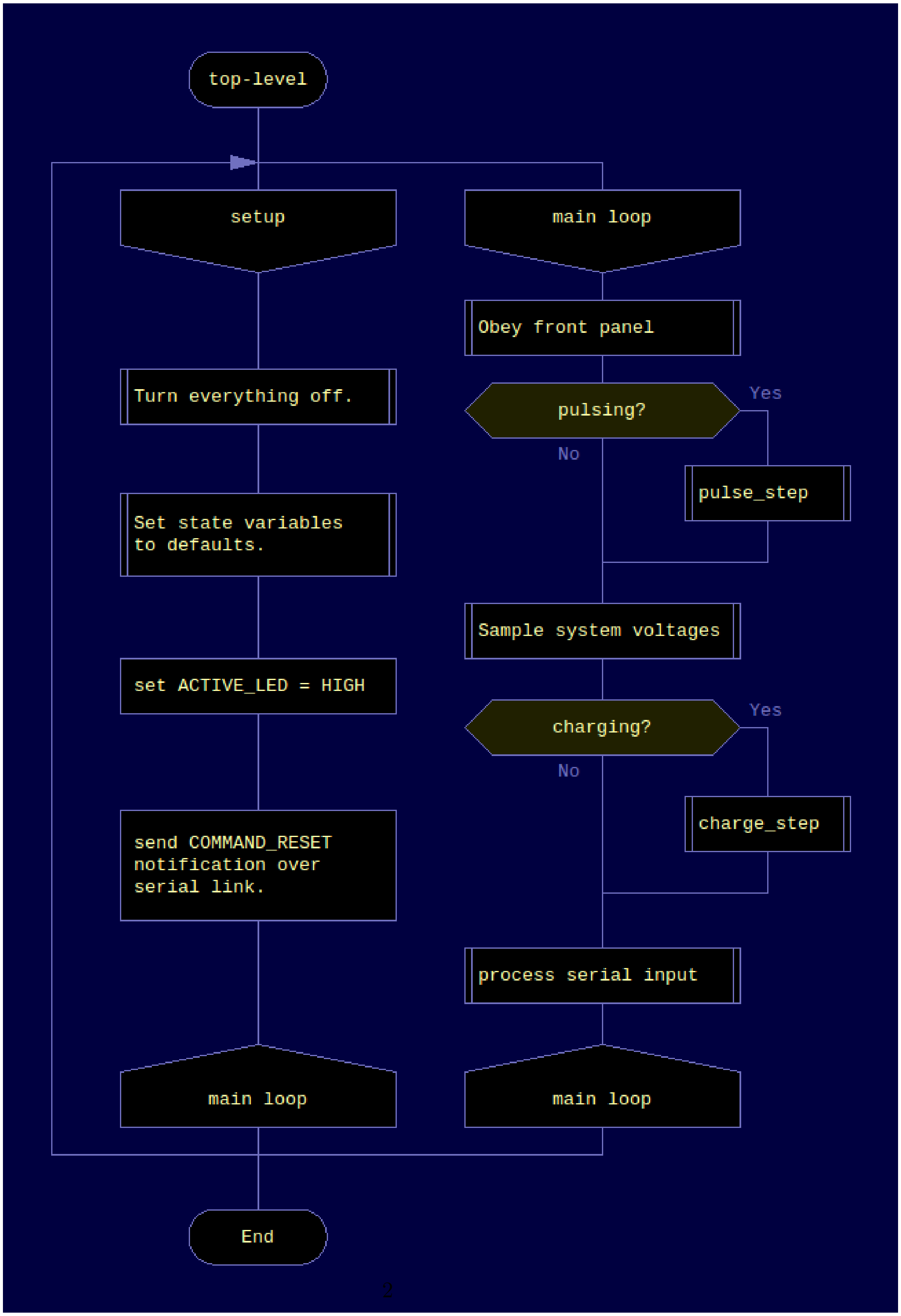

Figure 2: Top-level flow of the bank controller

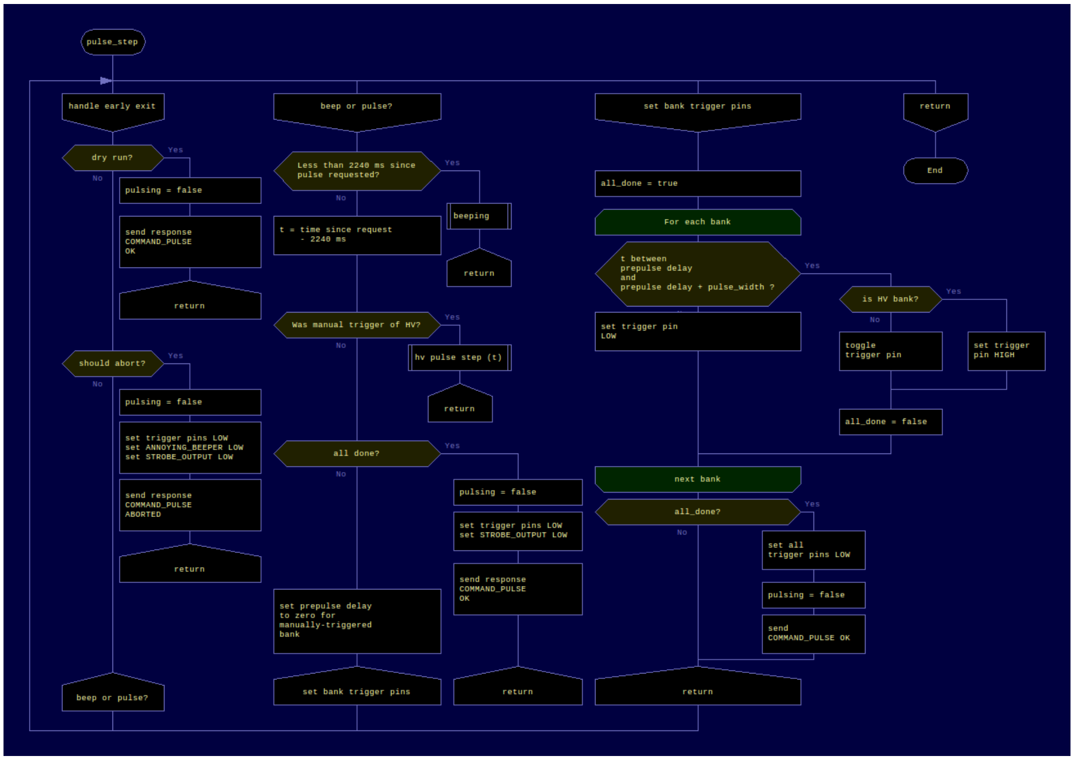

Figure 3: Pulse Step processing

The serial protocol used is nonstandard; a definition can be found at `https://github.com/hedj/nslc`. It is designed to facilitate debugging using a serial terminal; nslc is a simple frame-based protocol using line-feeds as end-of-frame markers and whitespace for escape. It is also checksummed for reliability. The particular checksum is only 8 bits in size, and is optimised for speed of implementation.

The most important thing to emphasise is that the controller is implemented as a tight loop reacting to the change in time. There are no blocking wait statements; in this way we ensure that input can continue to be received from the front panel switches or serial as the process runs. This allows the safe abort of the system regardless of its state, which is rather important.

The two operations which consume a reasonable amount of time and are thus implemented as conditional actions in the inner loop are pulsing and charging, respectively Figs. 3 and 4.

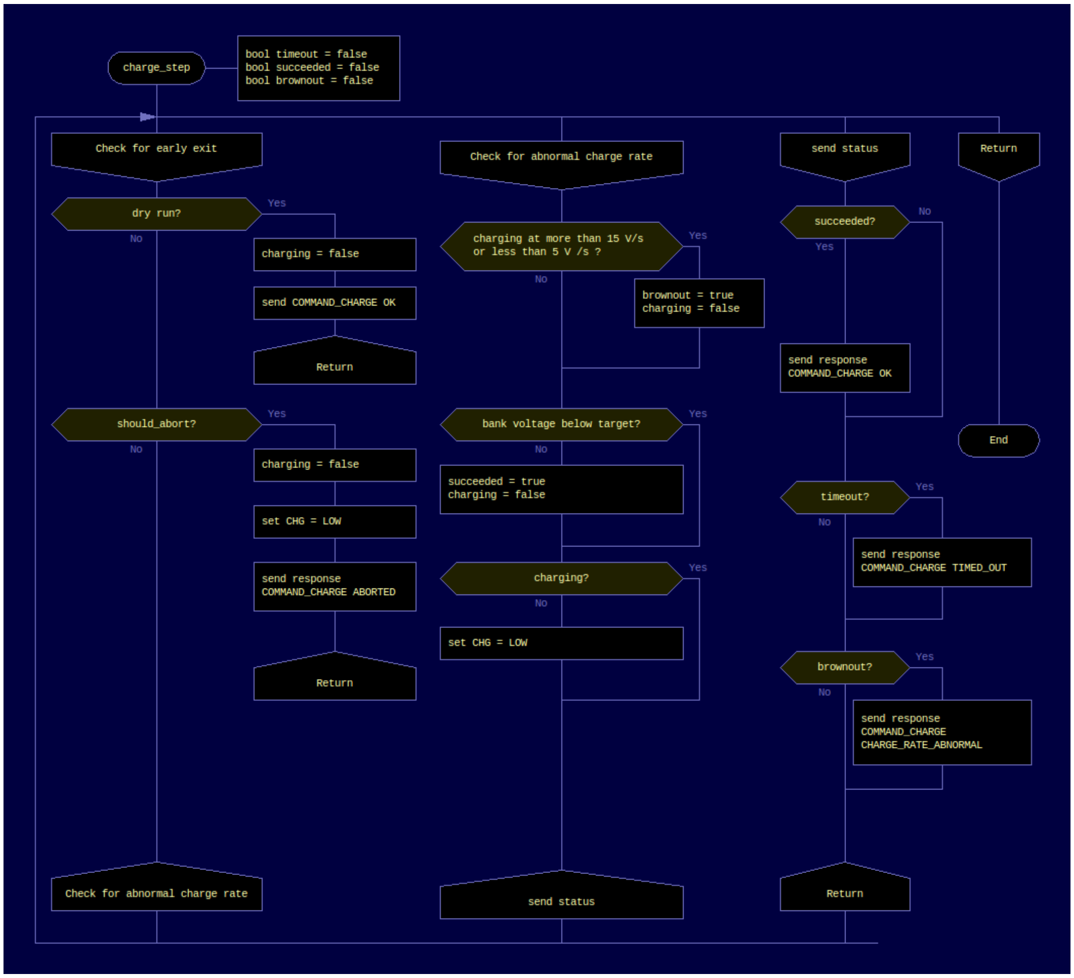

Figure 4: Charging step logic

## 1.2 The Stepper Controller

## 1.3 The IRC Server

## 1.4 The Agents

## 1.5 Using the System