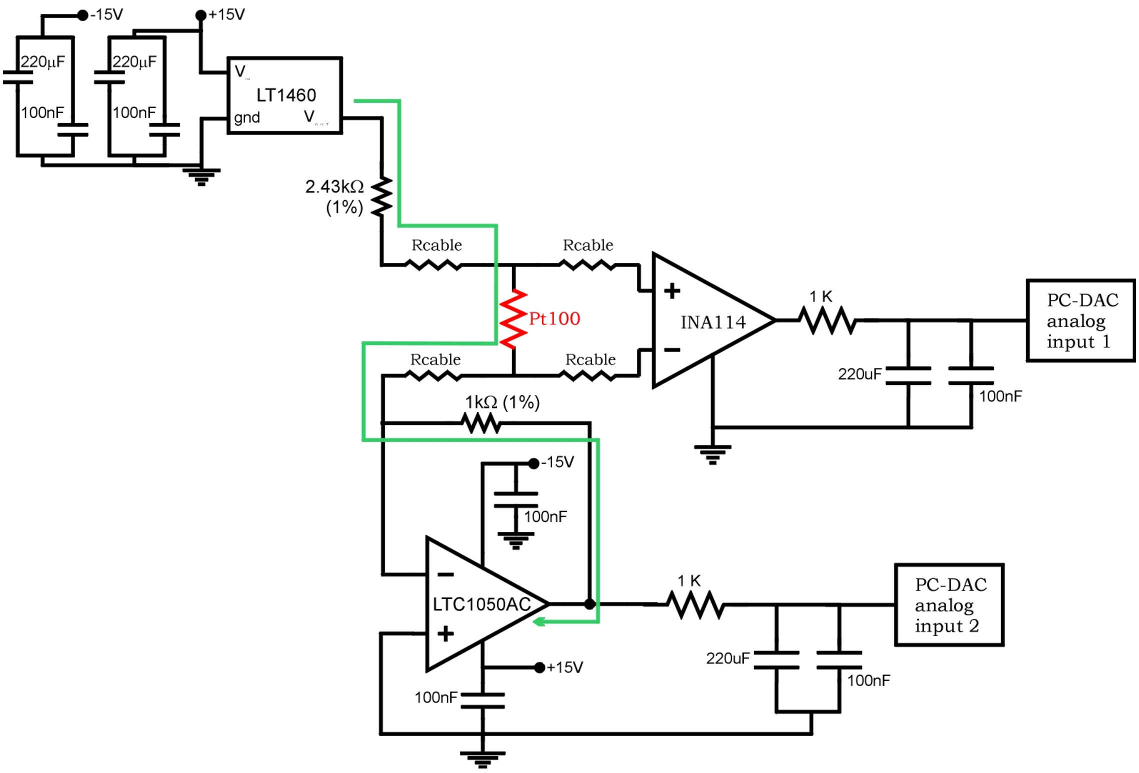

Figure 75 - Four-wire platinum resistance measurement circuit. The main current flow is shown in green.

In contrast to thermocouples, Pt100 resistances (platinum RTDs with a nominal 100 Ω resistance at 0 °C) are not self-powered thermoelectric devices. RTDs temperature sensing capabilities stem from the fact that platinum (and other materials) resistivity changes significantly (and in a quasi-linear manner) with temperature and this means that a current flow (illustrated in green) must be injected into the system to allow reading of the RTD resistance by Ohm's law. The injected current typically lies in the 0.1-1 mA range, because it should be as small as possible to avoid self-heating Joule effects on the PRTD, but also high enough to yield a significant signal and overcome noise. Additionally, a Pt100 can be read either in a 2-wire or a 4-wire configuration. The problem with 2-wire configurations is that the read resistance will be the overall resistance of the system (RTD plus cables), whilst in a 4-wire configuration current is injected through two independent wires, ensuring that the resistance read by the remaining two wires is purely the resistance of the RTD, since, theoretically, no appreciable amount of current is flowing across the reading wires.

In the present design, accurate temperature injection of 1 mA was accomplished using a LT1460-2.5 voltage reference source (*Linear Technology*) that gave a 2.5 V voltage output from the FAC-6628 ±15 V-1 A dual power supply (*Promax*) with 0.075 nominal accuracy. This output voltage was coupled to the Pt100 device via a 0.1% 2.43 kΩ resistor in a typical voltage divider scheme, meaning that the current passing through the Pt100 had to be approximately 2.5 V/(2430+100)Ω = 0.988 mA. To ensure that this was so, and to make accurate temperature readings, the current passing through the Pt100 was then read by a LTC1050 precision op-amp (*Linear Technology*) acting in a thousand-fold inverted-amplifier mode. The voltage drop at the Pt100 was measured with a separate set of cables by an INA114 precision differential amplifier (*Burr-Brown*) set at a ~10.107 gain by the 0.1% 5.49 kΩ resistor. The Pt100 circuitry readings were software-adjusted after calibration in a water bath with reference thermometers (see *Materials and Methods*, p.310). Once calibrated, Pt100 resistance could be easily deduced using the formulae in Equation 4 and temperature was then interpolated using a look-up-table to compensate for Pt100 non-linearities (see *Materials and Methods*, p.312).

$$R_{Pt100} = \frac{V_{Pt100}}{I_{Pt100}} \quad \Leftarrow \quad I_{Pt100} = -\frac{V_{DAC2}}{1000} \quad \& \quad V_{Pt100} = \frac{V_{DAC1}}{\sim 10}$$

Equation 4 - Pt100 resistance derived from differential and precision amplifier readings.

Circuitry assemblage

After development and fabrication (see *Materials and Methods*, p.303) of the aforementioned circuits, they were assembled to produce the basic operational electronics for the software PID control explained in next section. Although thermocouples were finally discarded in favor of Pt100 sensors, they were conserved, together with their acquisition circuitry, and integrated in the final assemblage to provide a contrast temperature measurement as a safety resort, and to yield information on room temperature. Thus, the final assemblage included a thermocouple sensing circuitry, a Pt100 sensing circuitry and a power driver. Two different power supplies were used for driving and sensing purposes and all systems were directly controlled by the PC data acquisition board via analog inputs/outputs through a 50-wire interface. The basic interconnection scheme of the final assemblage can be seen in Figure 76.

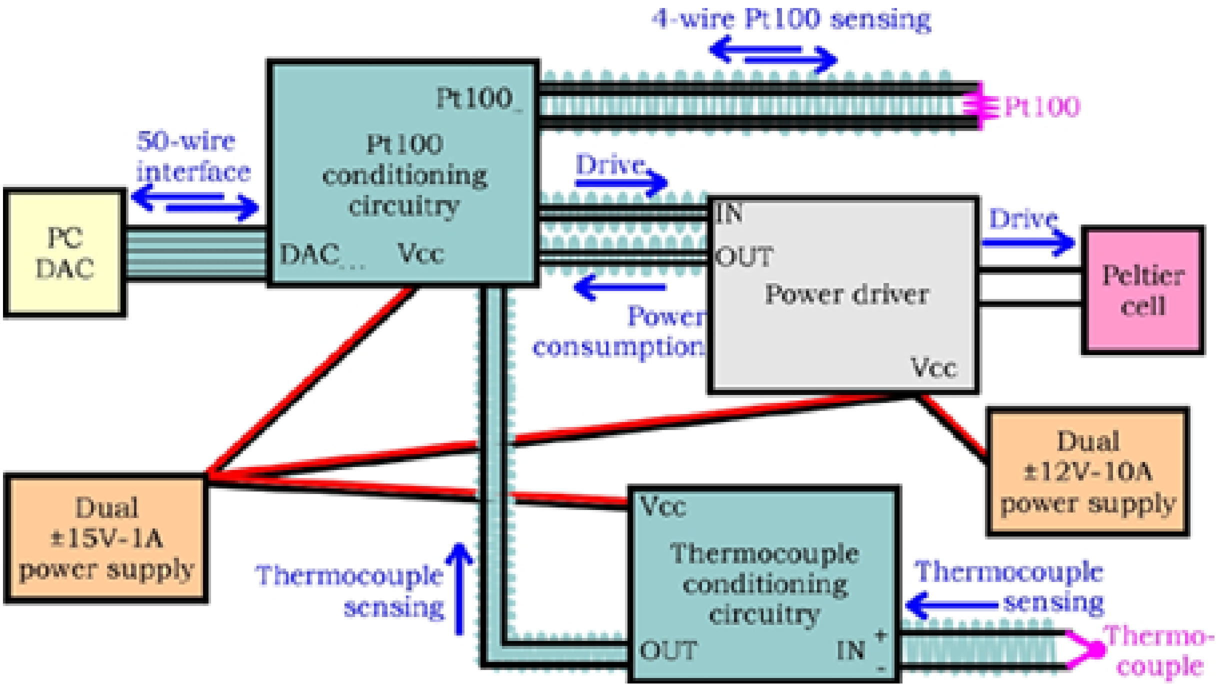

Figure 76 - Interconnection scheme of the final circuitry assemblage. Blue-green background zones stand for ground-shielded boxes and wires.

Since the Pt100 module was the last to be incorporated, it was also used as a bridge between the DAC interface and the other modules, and a high-speed analog multiplexer was added to it to allow multiplexing of the driver power consumption signals, because the number of DAC analog inputs was limited. To avoid the occurrence of ground loops that could easily interfere with the low-voltage measures of Pt100 readings, ground paths were distributed in a radial manner across all the systems. Also, to avoid RF interferences, all modules (except the driver module) were inserted in grounded metallic boxes and all signal paths were also shielded with a grounded metallic wire mesh.

4.4.5. SOFTWARE TEMPERATURE CONTROL

Software setup

Software platform

Temperature control was implemented with a Labview (*National Instruments*) program to allow easy fine-tuning of control parameters and the inclusion of different sequential step configurations (like system preheating, initial denaturation or the final fridge hold). Although programming could have been implemented in a number of different languages (or even loaded onto an external PLA (Programmable Logic Array) device, Labview was the best option available since it has a very flexible graphic interface, a large built-in set of instrument managing and data processing algorithms and, in this particular case, specific drivers for the DAC device used in this work: a PCI-1200 DAQ (*National Instruments*, see *Materials and Methods*, p.305). In addition, Labview programs can finally be compiled from their underlying G language and work efficiently on any Ms-Windows (*Microsoft*) based platform.

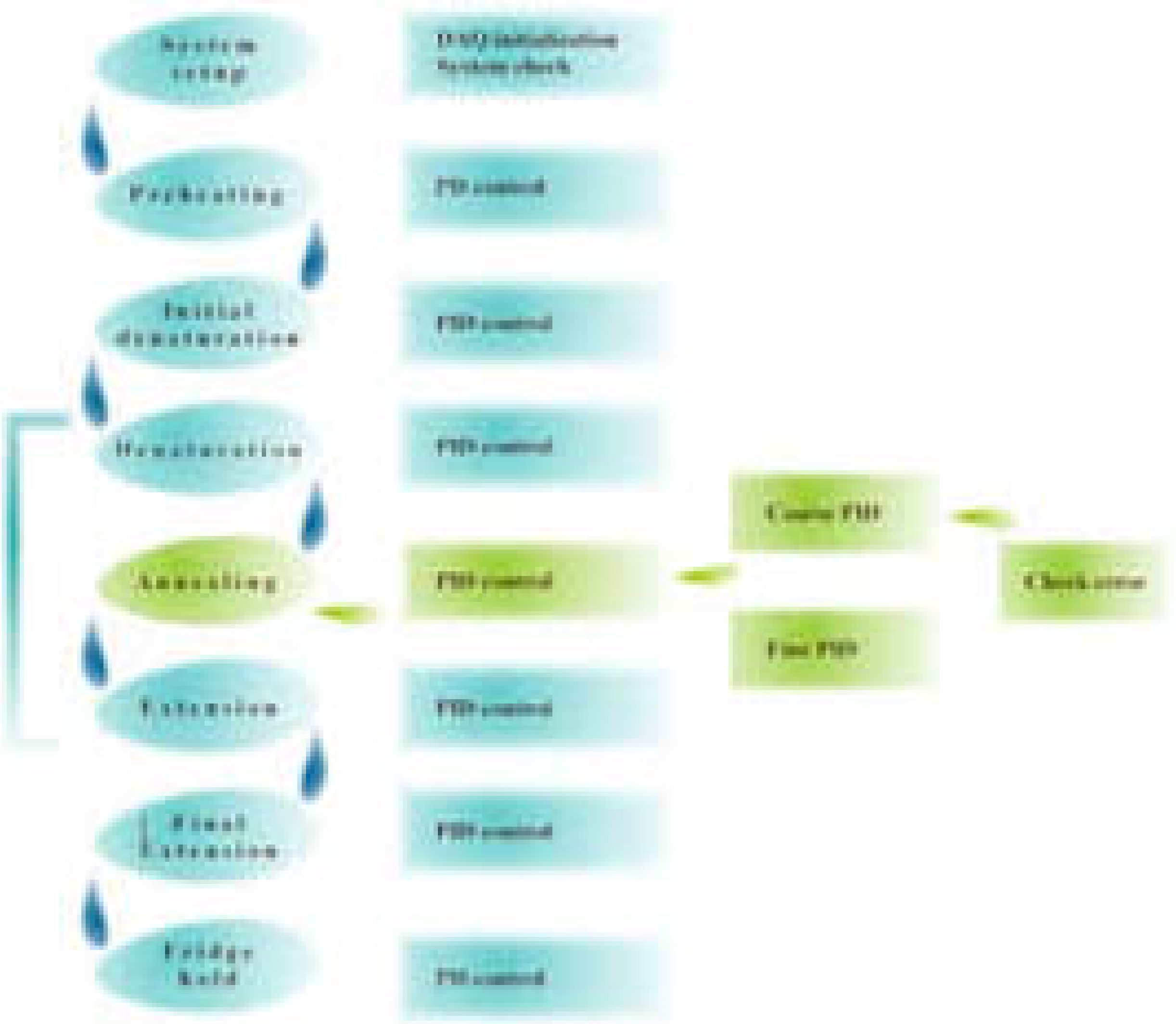

Figure 77 - Basic program structure, depicting the 8 different program sequences and their control modes, together with double-PID methodology, described later (see p.165).

The basic program structure can be seen in Figure 77. The program makes use of G *Sequence* structures to create a set of eight different program steps. First, a system setup (including DAC initialization and check-in procedures) is carried out. Afterwards, the system waits for the user to set the control parameters and a preheating step is carried out to allow hot-start PCR. Initial denaturation for a specified time interval ensues, and then the program enters the main PCR-cycling loop, where it provides denaturation, annealing and extension steps in a repetitive manner and for an specified number of cycles. After completing the main PCR phase, the program enters another sequence structure, providing a final extension period, and then goes off into a fridge hold loop until the user extracts the PCR-chip and terminates the program.

Functional units

Conceptually, the program is divided into four main functional levels. At the top level, the main sequence unit controls the heuristic PCR steps described above and the interaction with the user (input dialogs and display issues). At the second level lies the main PCR-loop, which is used in the central cycling stage of PCR and provides PCR-cycling by means of two nested FOR loops. Finally, at the third and fourth level lie the PID algorithm and the data acquisition board interface, which comprise by themselves two different functional units.

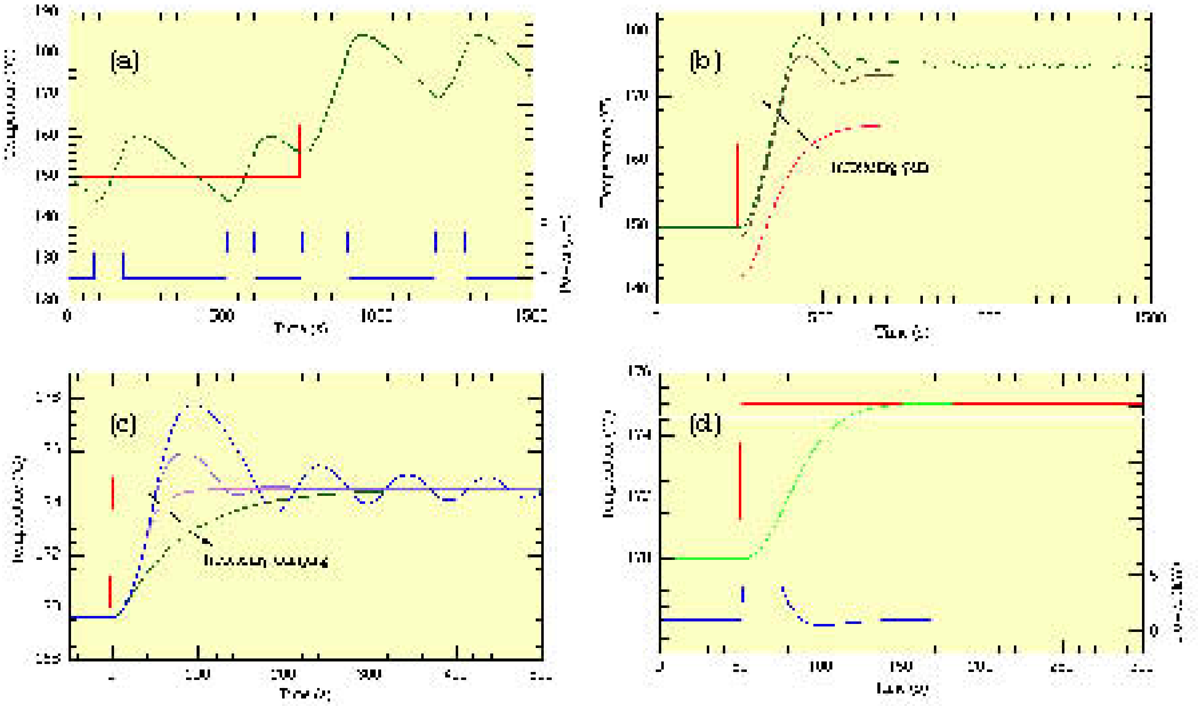

Figure 78 - Comparison of ON-OFF, proportional, proportional-derivative and proportional-integrative-derivative controls. Set temperatures are shown in red.