CAMBADA, Hardware Description

J. L. Azevedo, M. B. Cunha, A. J. R. Neves, N. Lau, A. Pereira,
G. Corrente, F. Santos, D. Martins, N. Figueiredo, J. Silva, J. Cunha,
B. Ribeiro, R. Sequeira, L. Almeida, L. S. Lopes, J. M. Rodrigues, A. J. Pinho

1. Introduction

The CAMBADA robots were designed and completely built in-house. Each robot is built upon a circular aluminum chassis (with roughly 485 mm diameter), which supports three independent motors (allowing for omnidirectional motion), an electromagnetic kicking device and three NiMH batteries. The remaining parts of the robot are placed in three higher layers, namely: the first layer upon the chassis is used to place all the electronic modules such as motor controllers; the second layer contains the PC (currently a 12" notebook based on an Intel Core2Duo processor); finally on the top of the robots stands an omnidirectional vision system based on a hyperbolic mirror (AIS Fraunhofer-Gesellschaft).

The mechanical structure of the robot is highly modular and was designed to facilitate maintenance. It is mainly composed of two tiers: 1) the "mechanical" section that includes the major mechanical parts attached to the aluminum plate (e.g. motors, kicker, batteries); 2) the "electronic" section that includes control modules, the PC and the vision system. These two sections can be easily separated from each other, allowing an easy access both to the mechanical components and to the electronic modules.

2. General Architecture of the Robots

The general architecture of the CAMBADA robots has been described in [1][2][3]. Basically, the robots architecture is centered on a main processing unit (a PC running the Linux operating system) that is responsible for the higher-level behavior coordination, i.e. the coordination layer. This main processing processes visual information gathered from the vision system, executes high-level control functions and handles external communication with the other robots. This unit also receives sensing information and sends actuating commands to control the robot behaviour by means of a distributed low level sensing/actuating system. The communication among team robots uses an adaptive TDMA transmission control protocol [6], on top of IEEE 802.11b, that reduces the probability of transmission collisions between team mates thus reducing the communication latency.

The low-level sensing/actuation system (Figure 1) is implemented through a set of microcontrollers interconnected by means of a CAN network (Controller Area Network (CAN) [7]). This network is complemented with a higher-level transmission control protocol to enhance its real-time performance, composability and fault-tolerance, namely the FTT-CAN protocol (Flexible Time-Triggered communication over CAN) [5]. The main blocks of the low-level sensing/actuation system are: motion control, odometry computation, compass, kicking control and system monitoring. The motion control block is composed of three independent motor control boards each of them receiving a velocity setpoint from the high-level holonomic motion controller. The odometry block combines the encoder readings from the 3 motors and provides coherent robot displacement information that is periodically sent to the high level coordination layer. The compass block reads the compass sensor and sends periodically to the high-level the corresponding read value. The kicking control block includes the control of an electromagnetic kicker and of a ball handler to dribble the ball. Finally, the system monitor, which is in fact a distributed function, monitors the robot batteries as well as the state of all nodes in the low-level layer.

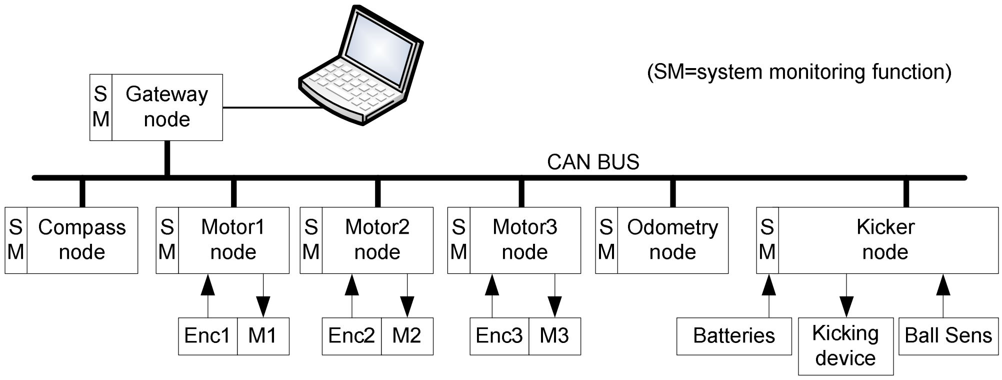

Figure 1. The CAMBADA low-level hardware architecture.

The low-level control layer connects to the coordination layer through a gateway (see Figure 2 for the electrical drawing of this module), which filters interactions within both layers, passing through the information that is relevant across the layers, only.

3. Low-level hardware description

The low-level layer nodes are interconnected with a CAN network operating at a bit rate of 250Kbps. A gateway interconnects the CAN network to the PC at the high-level layer either through a serial port or a USB port, operating at 115 Kbps in any case. All modules (except for the motor control modules) are based on the same underlying hardware, e.g. a PIC18Fxx8 Microchip [4] microcontroller (@40MHz, i.e., 10 MIPS) which, along with a set of useful peripherals, such as timers, PWM generators, analog to digital converter and serial communications, also integrates a CAN controller. The basic structure of every module includes the CAN port to connect to the network and also includes a 115 Kbps RS232 serial port, which is useful both to program the module firmware (through a boot-loader) and for debugging purposes.

One important characteristic of the CAMBADA hardware design is the galvanic decoupling between the "logic" blocks (e.g. microcontrollers, PC, cameras) and the "power" blocks (e.g. motors, kicker) carried out through optocouplers and isolation amplifiers. Along with improved reliability of the whole system it prevents serious damages in expensive equipment (such as the notebook in the high-level layer) whenever any electric problem occurs in the "power" block. The drawback of this solution is the need of an extra battery for the "logic" part of the system. Thus, the low-level hardware modules are powered through 3 NiMh batteries: one 9.6V, 3700 mAh for the "logic" blocks and two 12V, 3700 mAh for the "power" blocks.

The following sections describe in more detail each node of the low-level sensing/actuation system.

3.1. Motion control

The robot holonomic motion is obtained combining the speed of 3 DC motors (Maxon 24V-150W), each with its own speed controller. Each of these controllers is a distinct module of the whole distributed architecture implementing a PI closed loop speed control. It takes as inputs the motor shaft displacement, obtained through a quadrature 500 P/R incremental optical shaft encoder coupled to the main axis of the motor, and the speed setpoint. The hardware of these modules has two main blocks: 1) the "logic" block (based on a Microchip dsPIC) that interfaces to the rest of the system and generates the required control signals, and 2) the "power" block which is essentially an NMOS H-Bridge, with two high-side drivers, to actually drive the motor. The output of the *logic* block is a set of two 20 KHz PWM signals implementing a modified lock anti-phase drive. In this drive mode the motor is energized only during the on-time, in contrast with the standard lock anti-phase where the motor is energized in reverse direction during the off-time. That is, when the motor is stopped (duty-cycle of the PWM signals is 50%) the current is zero. This implementation leads to a significant gain in battery autonomy, whenever the motor is not rotating at its maximum speed. Figure 3 and Figure 4 present the complete electrical drawing for this module.

3.2. Odometry

The odometry function of the robot is accomplished through the combination of 2 basic functions: the reading of each one of the 3 encoders plus their combination to generate coherent robot displacement information (Δx, Δy, $\Delta \theta$). The reading of each encoder is naturally allocated to each motor control module, using the same readings as those used by the speed feedback control. The combination of the readings is carried out in a specific module, the *odometry node*, which receives the encoder readings from the motor control modules and sends the results to the gateway (and to the high-level control) via CAN messages. This module is implemented with a Microchip PIC18F258 microcontroller, as presented in Figure 5.

3.3. Kicking control

The kicking system, which has been fully developed by the CAMBADA team, is the so-called electromagnetic kicker. The main element of it is an electromechanical solenoid which consists of a coil wound around a movable iron core producing a magnetic field when an electric current passes through it. The magnetic field causes the iron core to move towards the ball, thus kicking it. Controlling the magnetic field provides control over the kicking power, and that represents a very convenient way to modulate the kicking action. The energy needed to drive the solenoid is stored in a capacitor. To get a strong kick a large magnetic field has to be created which implies the usage of reasonably high currents and/or voltages and also of large capacitors.

The kicking system is based on a Microchip PIC18F458. It follows the same basic structure already presented for the motor controller, that is, galvanic decoupling between the "logic" block and the "power" block. The two main components of the "power" block are: 1) a DC to DC converter circuit that stores energy in a capacitor; 2) a solid-state switch that controls the discharge of the capacitor on the solenoid thus triggering the kicker.

The DC to DC converter is a typical switch-mode converter based on a boost configuration that converts 24V DC to 90V DC. In general terms, it works in two steps:

1) a DC voltage is set across an inductor during a pre-defined period of time which causes the inductor to store energy magnetically; 2) the voltage is switched off which causes the stored energy to be transferred to the capacitor. Although very simple, this circuit is very efficient, resulting in a rather low capacitor recharge time. The implemented circuit works at 18KHz and, in practical terms, the recharge (from 24V to 90V) of a 80000µF capacitor takes roughly 6s. The capacitor charging process is carried out in a closed-loop way, being the voltage across the capacitor continuously monitored by the microcontroller. The output of the microcontroller is a 18 KHz / 35% PWM signal which has been found experimentally as optimal in order to minimize the charging time. This is crucial since an inefficient charging process can dramatically decrease the running time of the battery.

The second component of the above referred "power" block is a solid-state switch based on NMOS transistors, whose specifications (150V / 240A in our design) depend essentially on the capacitor voltage and current drawn by the solenoid.

The kicking system also includes an IR barrier which is used to detect the ball when it is in the kicking position, thus avoiding false triggering.

Another feature implemented in this module is an active ball-handler system whose purpose is to dribble the ball throughout the game field.

The functions related to the kicking system are executed within the kicker module, without need for additional modules. The kicker interacts directly with the high-level layer through the gateway via CAN messages. The complete electrical drawing of the kicking controller is presented in Figure 6, Figure 7 and Figure 8.

3.4. System monitoring

This functionality has two main purposes: measure batteries voltage and monitor modules run-time status. The latter requires this function to be present in all modules, tracking reset situations, namely power-up reset, warm reset, brown-out reset (caused by undervoltage spikes) and watchdog reset, as well as answering to *I'm alive* requests issued by the high-level control layer. Battery voltage monitoring is implemented in the same module as the kicker, since it already includes specific voltage monitoring hardware. The battery monitoring function continuously measures the voltage of the three NiMH batteries used in the robot, namely 2x12V for the "power" blocks of motor controllers and kicker, plus a 9.6V for the "logic" blocks.

The information gathered by the system monitoring function, in all nodes, is periodically sent to the high-level layer for remote monitoring purposes.

3.5. Compass module

The compass module (see Figure 9 for the electrical drawing) is based on a Microchip PIC18F258 microcontroller and on the Hitachi HM55B, a 2-axis digital integrated magnetic field sensor. The connection between the microcontroller and the compass sensor is accomplished through a synchronous serial protocol which enables full operation of the sensor.

This module reads periodically the compass sensor and sends the read data to the high-level modules via CAN messages.

4. Electrical drawings

4.1. Gateway

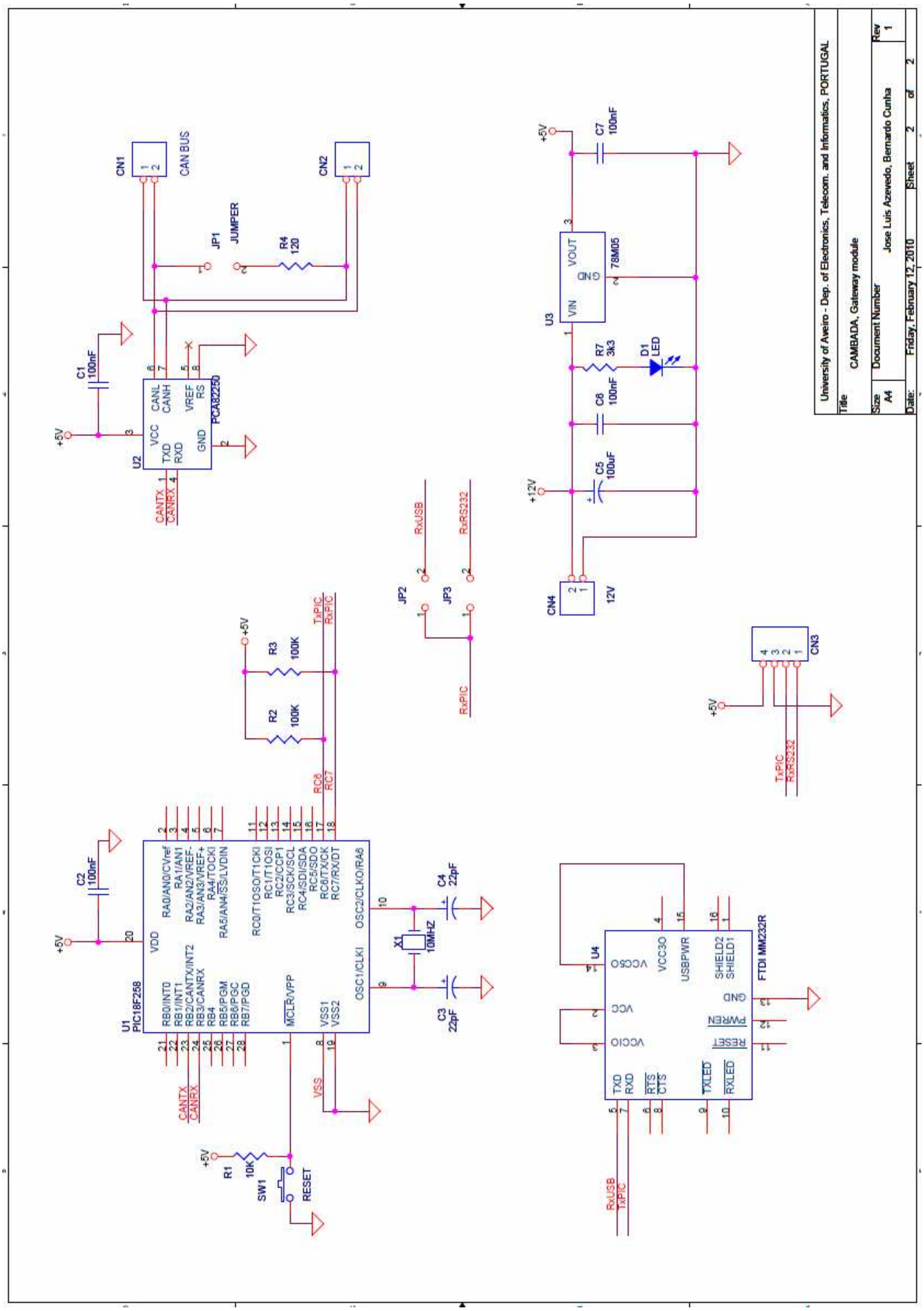

Figure 2. Gateway module.

4.2. Motor control

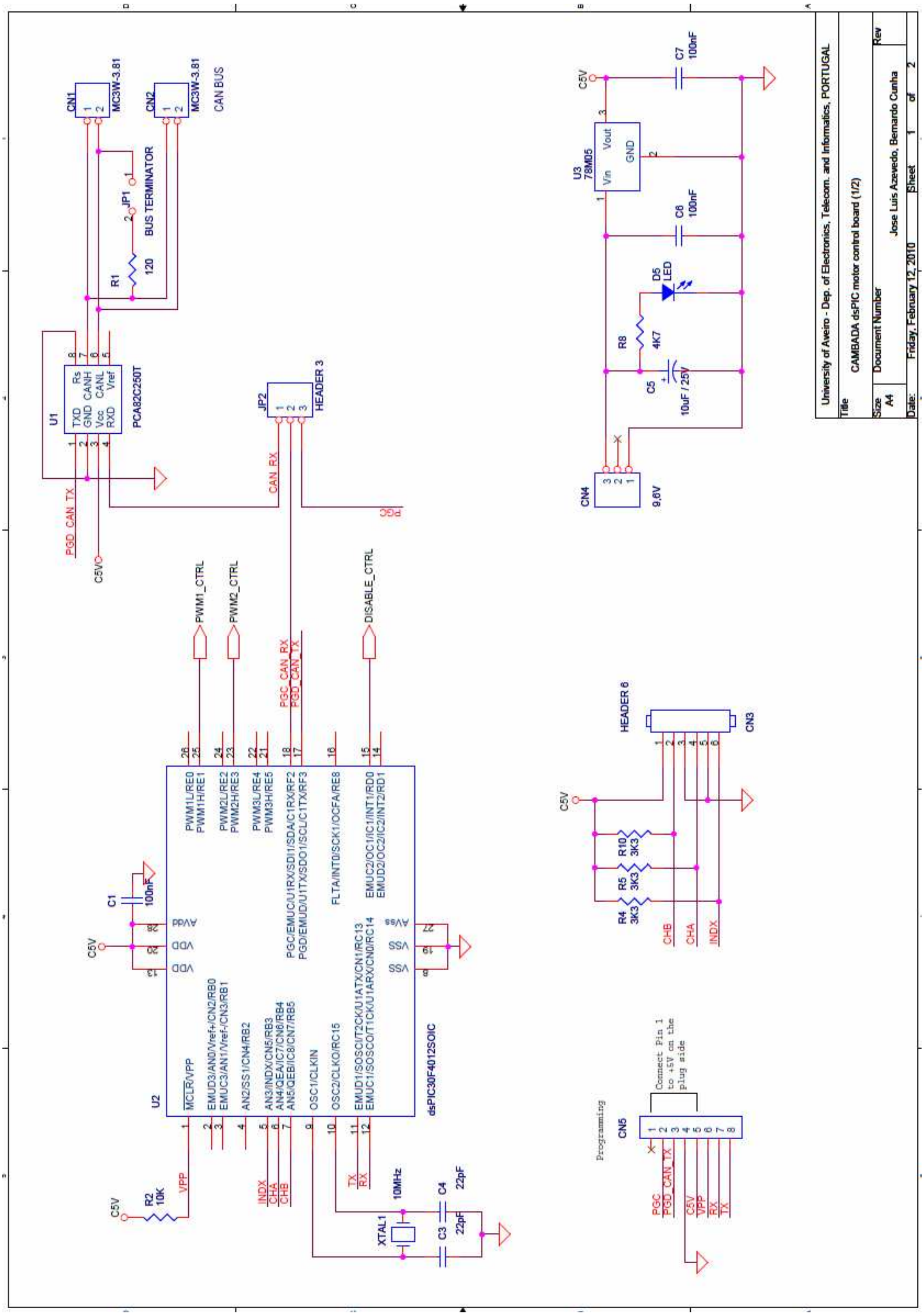

Figure 3. Motor control module (part 1).

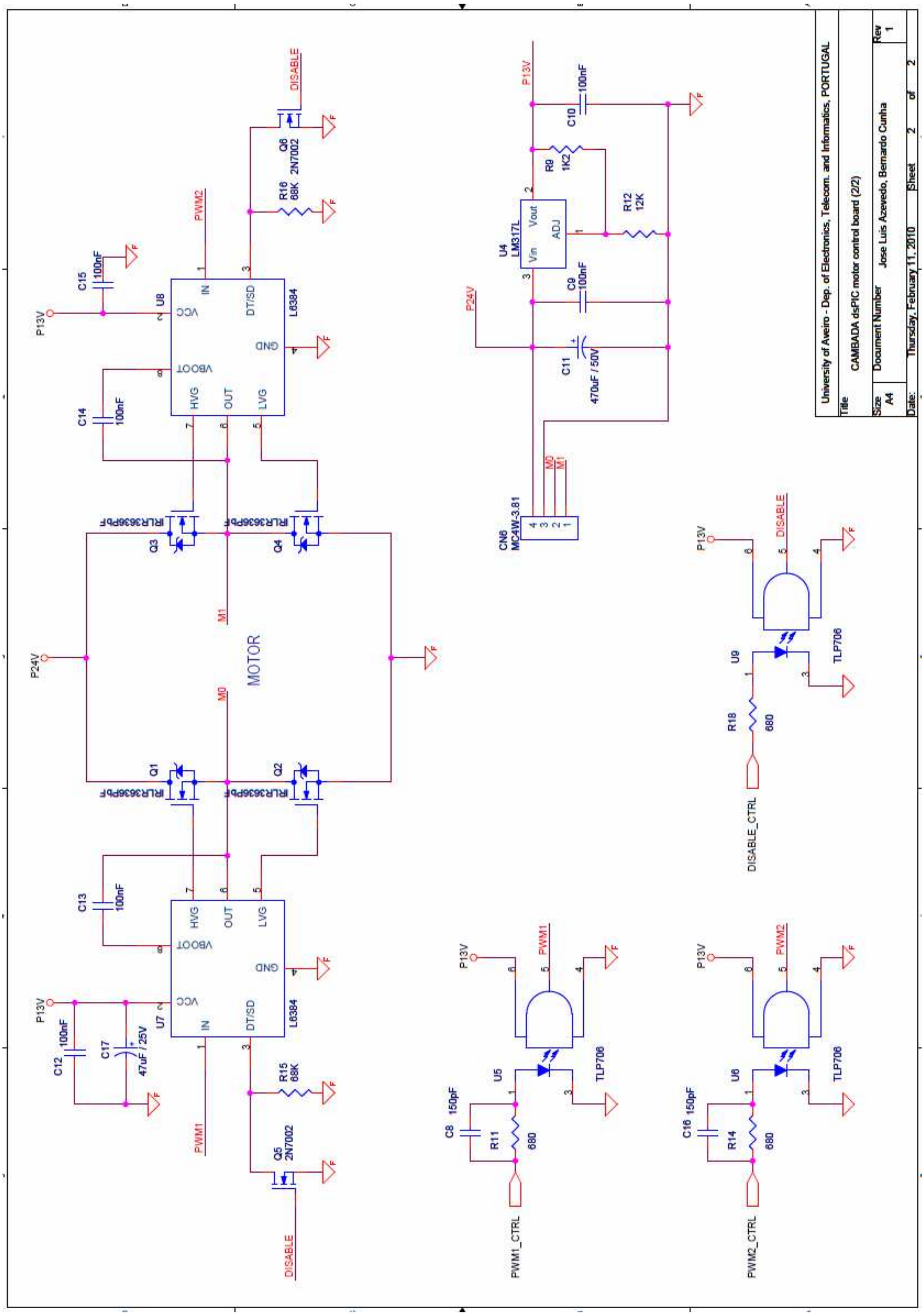

Figure 4. Motor control module (part 2).

4.3. Odometry

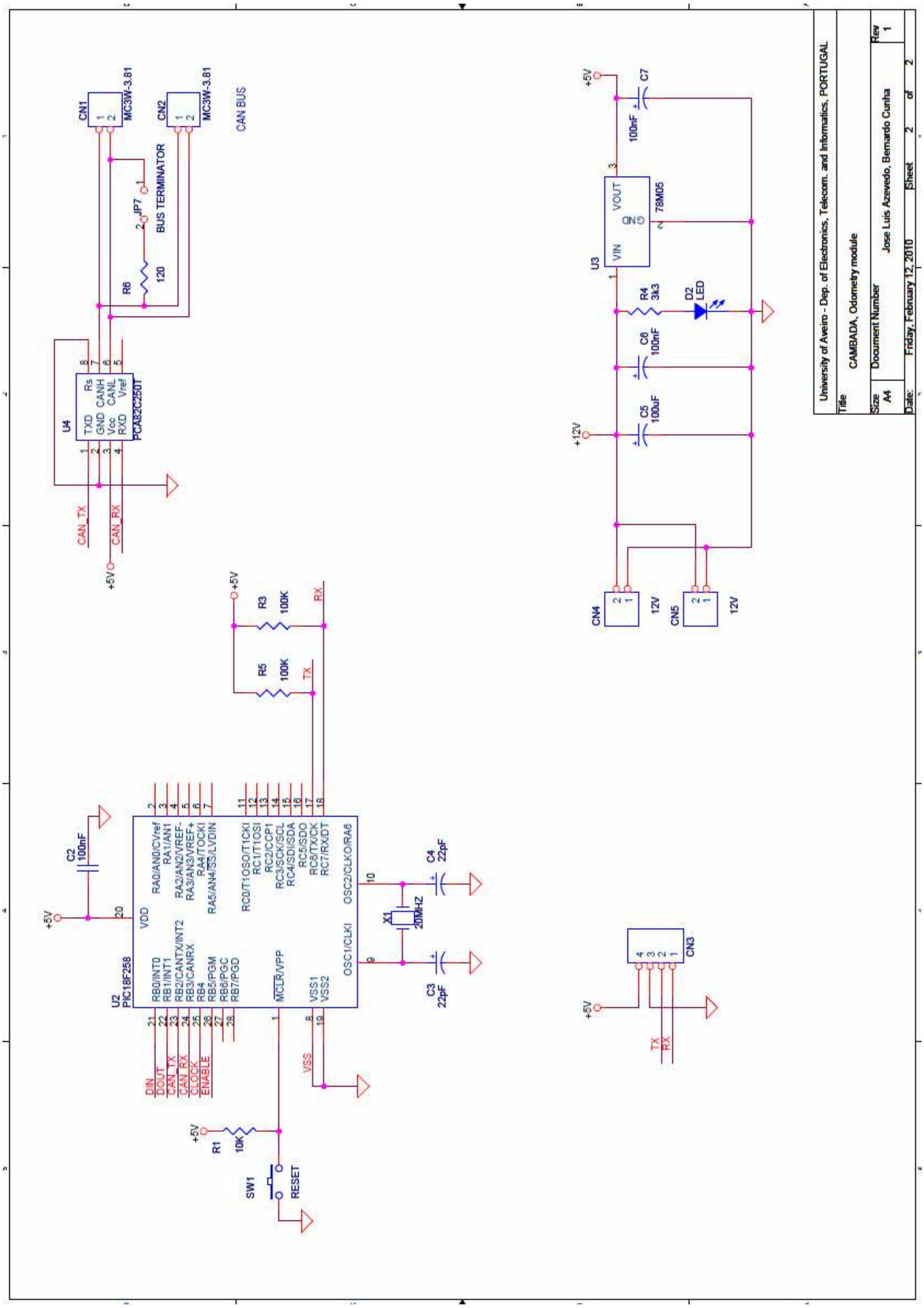

Figure 5. Odometry module.

4.4. Kicking control

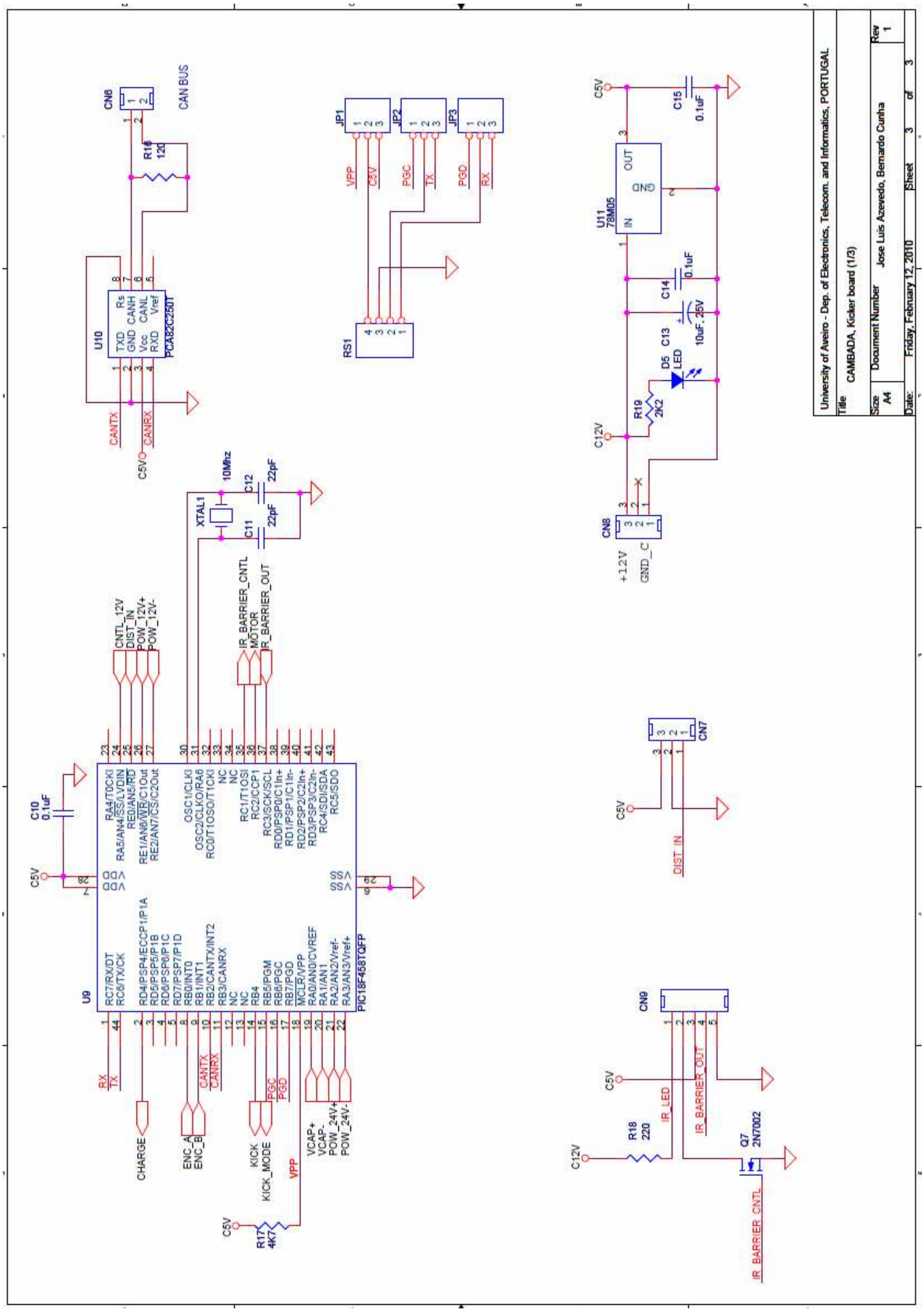

Figure 6. Kicking control module (part 1).

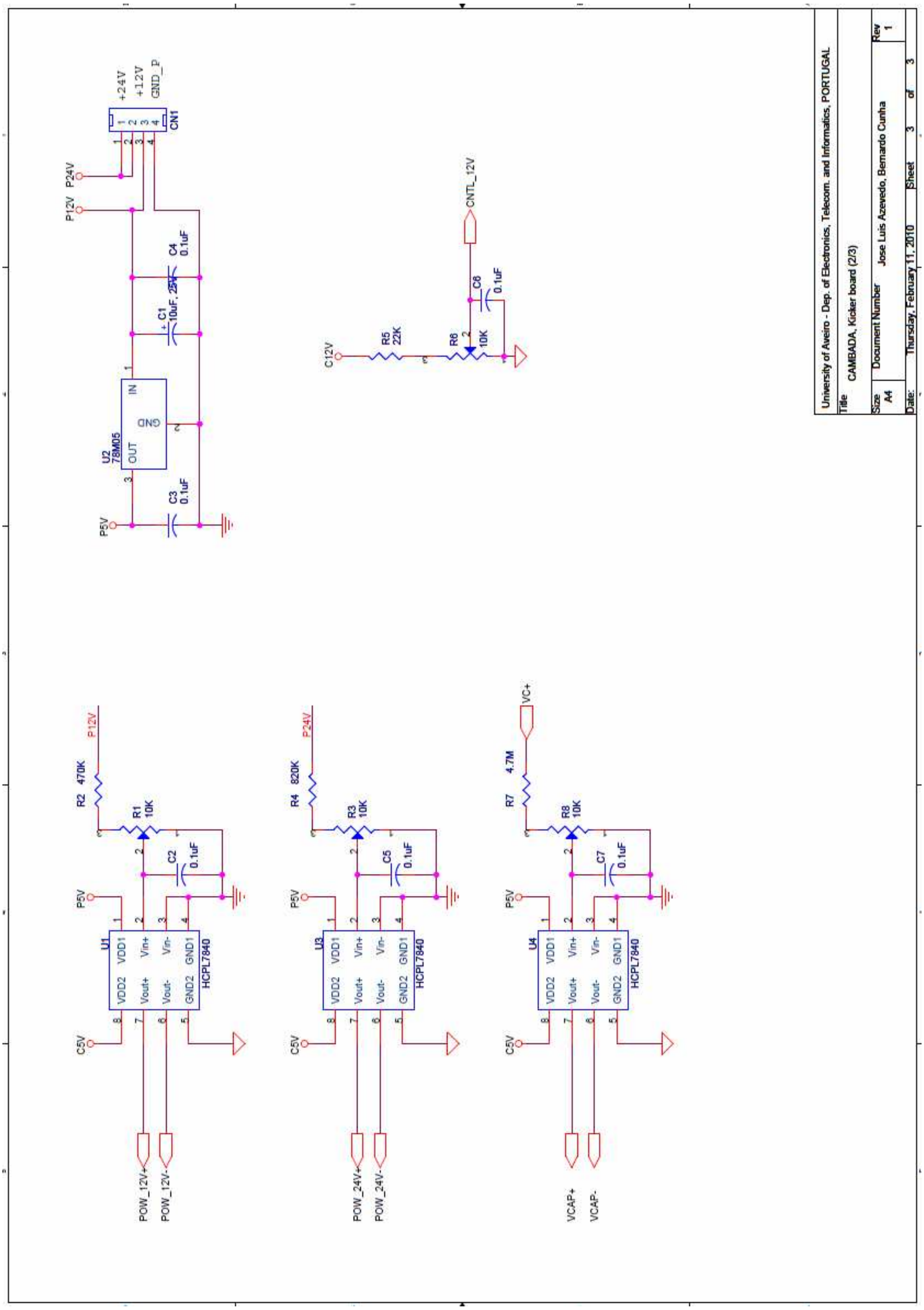

Figure 7. Kicking control module (part 2).

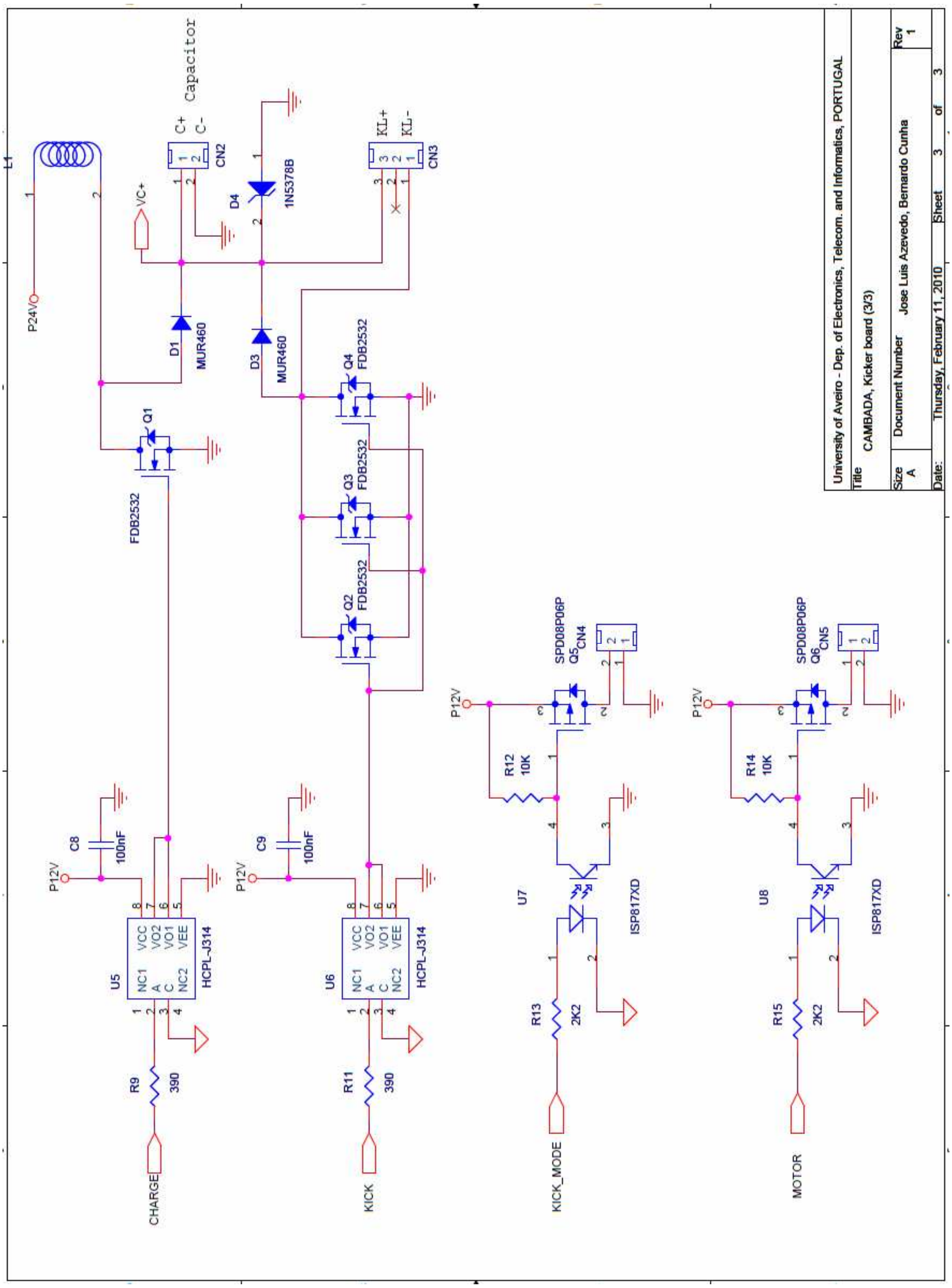

Figure 8. Kicking control module (part 3).

4.5. Compass module

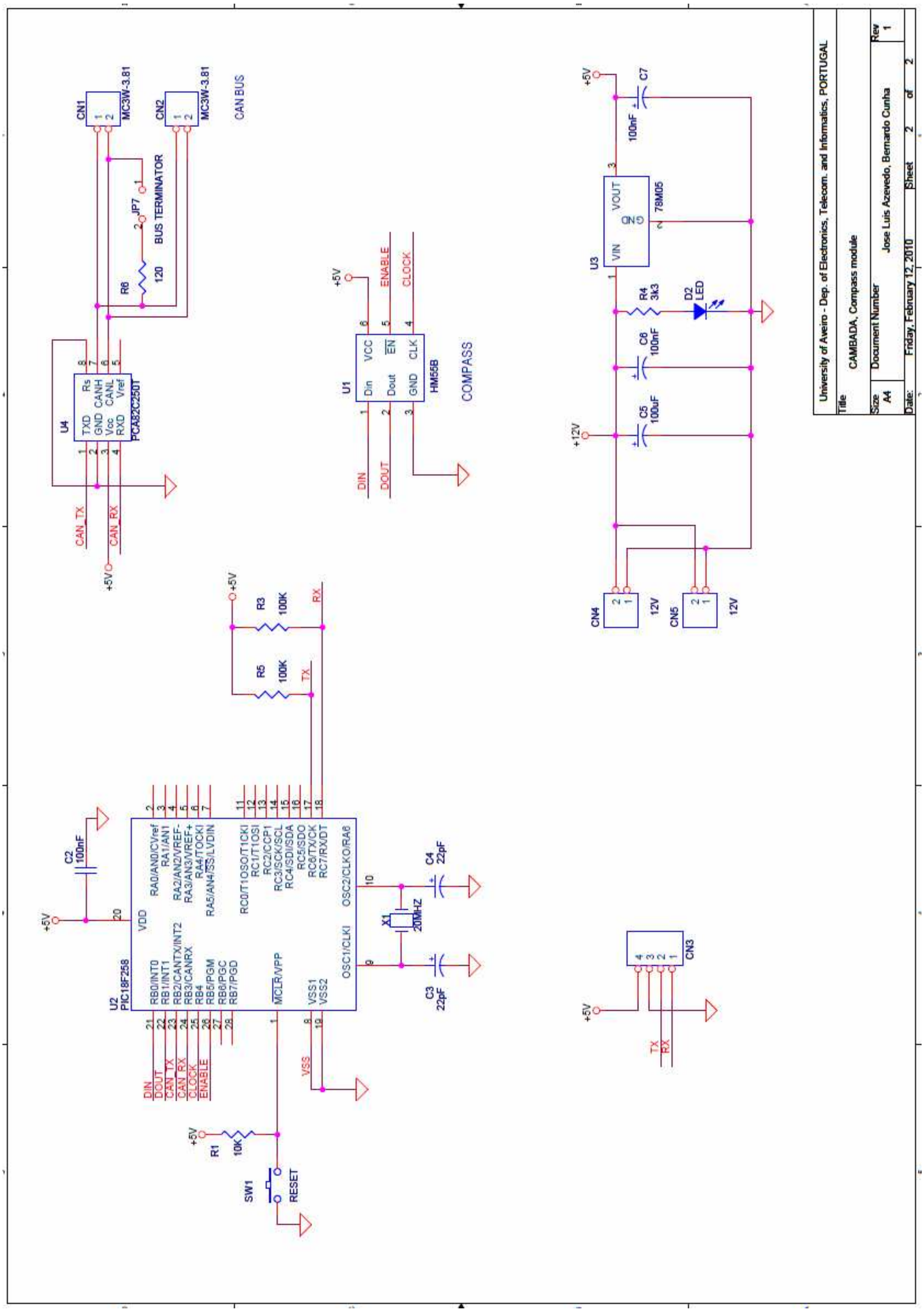

Figure 9. Compass module.

5. References

[1] L. Almeida, F. Santos, T. Facchinetti, P. Pedreiras, V. Silva, L.S.Lopes, "Coordinating distributed autonomous agents with a real-time database: The CAMBADA project". ISCIS'04, 19th International Symposium on Computer and Information Sciences. 27-29 October 2004, Kemer - Antalya, Turkey.

[2] V. Silva, R. Marau, L. Almeida, J. Ferreira, M. Calha, P. Pedreiras, J. Fonseca, "Implementing a distributed sensing and actuation system: The CAMBADA robots case study", IEEE ETFA 2005, Catania, Italy. September 2005.

[3] José Luís Azevedo, Manuel Bernardo Cunha, Luís Almeida (2007), Hierarchical Distributed Architectures for Autonomous Mobile Robots: a Case Study. ETFA2007- 12th IEEE Conference on Emerging Technologies and Factory Automation, pp. 973 - 980, Patras (Grece) September 25-28, 2007.

[4] Microchip website, available at www.microchip.com

[5] Almeida L., P. Pedreiras, J. A. Fonseca, "The FTT-CAN Protocol: Why and How, IEEE Transactions on Industrial Electronics", 49(6), December 2002.

[6] F. Santos, L. Almeida, P. Pedreiras, L.S.Lopes, T. Facchinnetti, "An Adaptive TDMA Protocol for Soft Real-Time Wireless Communication Among Mobile Computing Agents". WACERTS 2004, Workshop on Architectures for Cooperative Embedded Real-Time Systems (satellite of RTSS 2004). Lisboa, Portugal, 5-8 Dec. 2004.

[7] Controller Area Network - CAN2.0, Technical Specification, Robert Bosch, 1992.