ML-VPN

OtomaDUINO

Demo Examples

DISCLAIMER
Bortek provides this document “as is”, without warranty of any kind. Bortek may make improvements and/or changes in this document or in the product described in this document at any time. This document could include technical inaccuracies or typographical errors.

The user is solely responsible for all respects of design, development and production for his/her own products, including but not limited to: designing final products or components; verifying and testing such final products or components under actual operating conditions and applicable circumstances; determining the appropriateness of the use of Information in such final products or components; evaluating and determining the applicability of any information.

TRADEMARKS
Bortek and OtomaDUINO are trademarks of Bortek Boron Technologies and Mechatronics Inc. Other names mentioned in this document are trademarks/registered trademarks of their respective owners.

USING THIS DOCUMENT
This document is intended for the end user’s reference and provides brief usage information.

Though every effort has been made to ensure that this document is current and accurate, more information may have become available subsequent to the production of this guide.

REVISION HISTORY

Revision	Release Date	Release Description
1.0	04/10/2022	Initial release

SYMBOLS

Symbol	Description
 *Danger*	Indicates a hazardous situation which, if not avoided, will or could result in equipment damage.
 *Caution*	Indicates a potentially hazardous situation which, if not avoided, could result in data loss, performance degradation, or unexpected results.
 *Note*	Provides additional information to emphasize or supplement important points of the main text.

SAFETY INSTRUCTIONS

- The power must be disconnected before wiring the device.
- Wiring must be carried out in accordance with the device specifications. Otherwise the device will be damaged or unable to function.
- Do not touch the module terminals when the terminals have power.
- Do not connect or disconnect any other component when the device is power.
- DC sources must be isolated from the main AC power supply.
- Do not use joint power with continuous load or the input circuit of the controller.
- The device must be at least 50 mm in distance to the panel surface where it is installed and adequate ventilation conditions must be met.
- Ambient conditions should not exceed the temperature and humidity limits specified in the technical specifications.
- The device is CE marked according to EN standards.

ABBREVIATIONS

MCU	Micro Controller Unit
OtD	OtomaDUINO
PTO	Pulse Train Output
PWM	Pulse Width Modulation
Ch	Channel

Contents

1 Overview

This document provides information on the examples of OtomaDUINO.

2 System Requirements

- PC with Arduino IDE (1.6.4 or newer) (Windows, Linux, MAC)
- Internet connection
- USB Cable
- 12 or 24VDC (minimum) 20W power supply

3 Description

OtomaDUINO is an Arduino IDE compatible device with digital I/O, analog input and pulse output. The hardware consists of an Atmel microcontroller with a bootloader allowing quick in-circuit firmware updates.

OtomaDUINO was designed for industrial environments. It has DIN rail mount and all of the terminal blocks are spring type for fast deployment.

The demo examples are designed for utilization of the main functions of OtomaDUINO. Examples can be combined or changed the specific purpose.

The demo examples are provided as it is. The application of demo examples are in responsibility of the user.

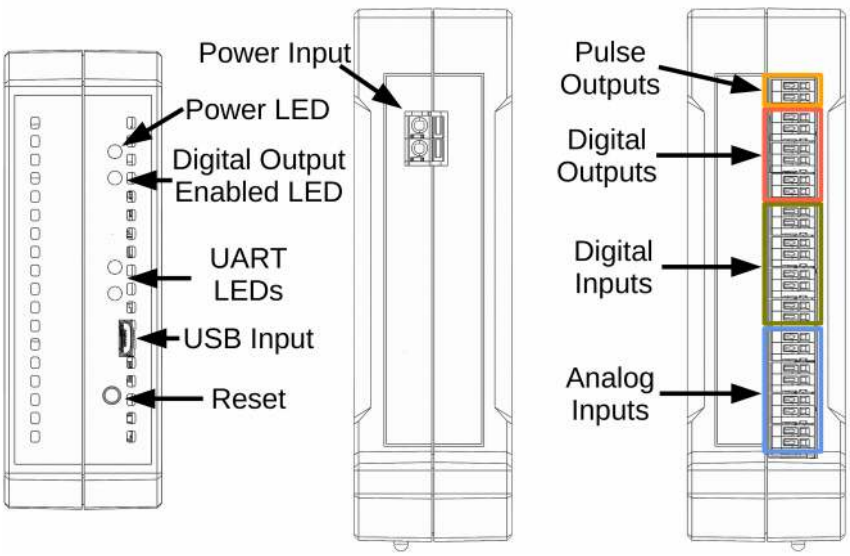

Figure 1. Front, bottom and top views of OtomaDUINO

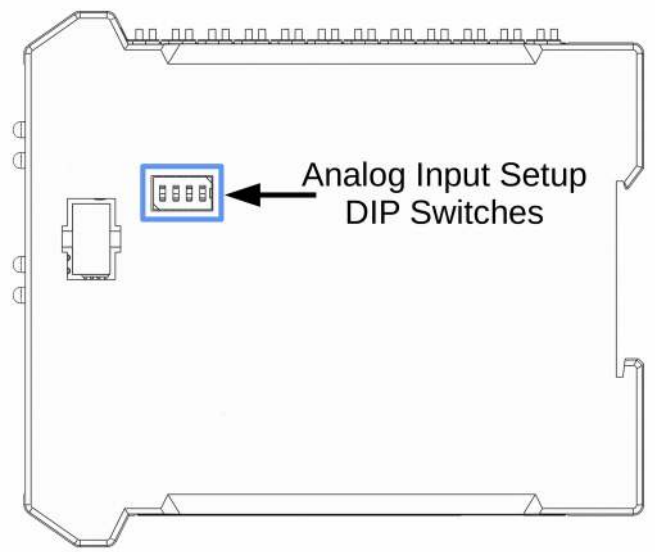

Figure 2. The side view of OtomaDUINO

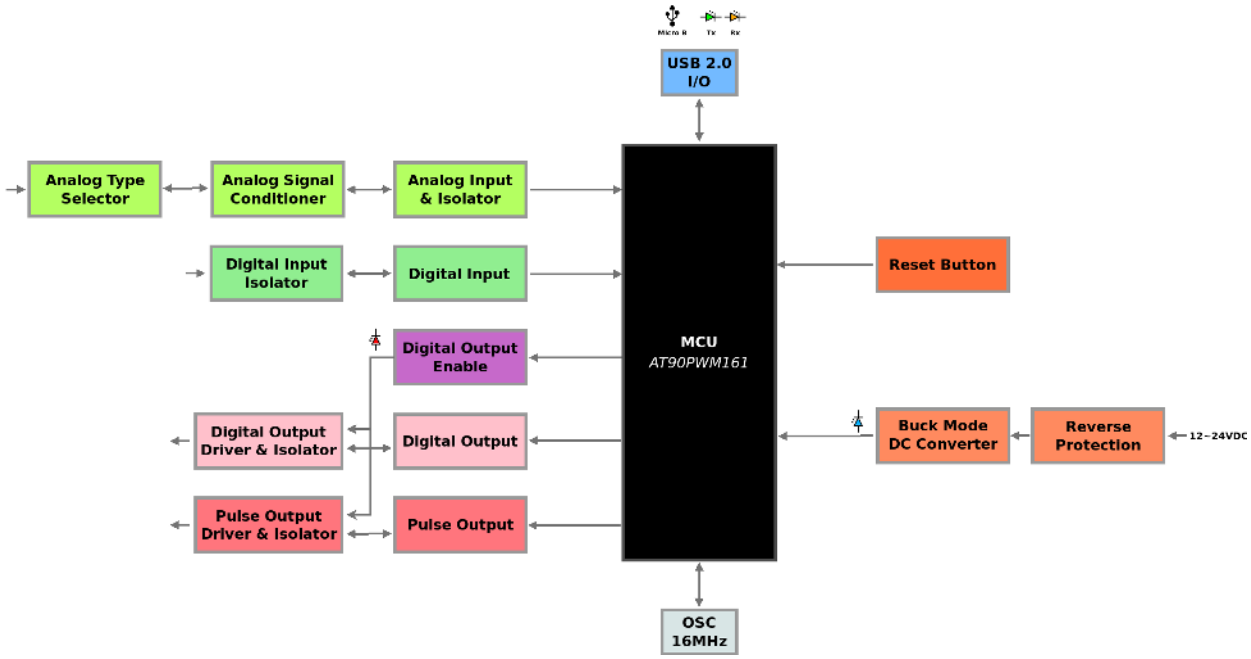

Figure 3. The block diagram of OtomaDUINO

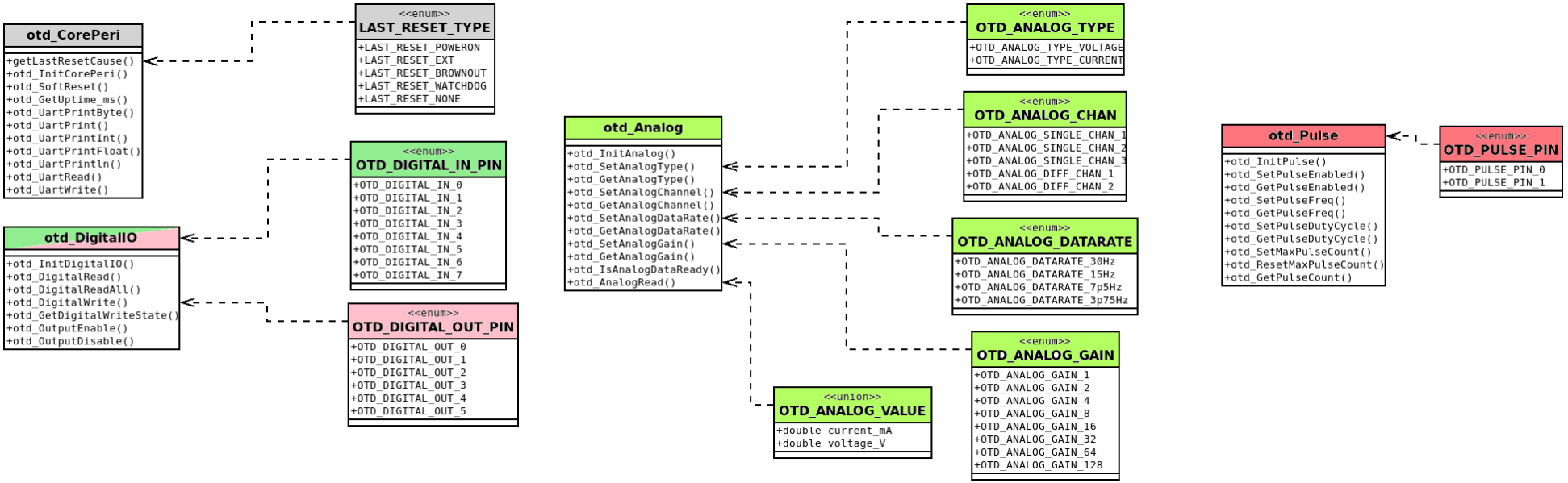

Figure 4. OtD Library Overview

4 Examples

The **power** must be **disconnected or completely turned off** before wiring the devices.

4.1 Hello World

This examples utilizes the usb serial output feature and software reset. In order to view the "**Hello World**" output, open the Arduino IDE Serial Monitor or any other terminal communication program (such as Hyperterminal, Putty, CuteCom etc) and use the following settings:

Baud rate = 19200
Data = 8 bits
Parity = None
Stop = 1 bit
Flow Control = None

Software Description:

Software basically initializes core peripherals and sends "Hello World" message over USB to Serial converter, waits 2 seconds and applies soft reset.

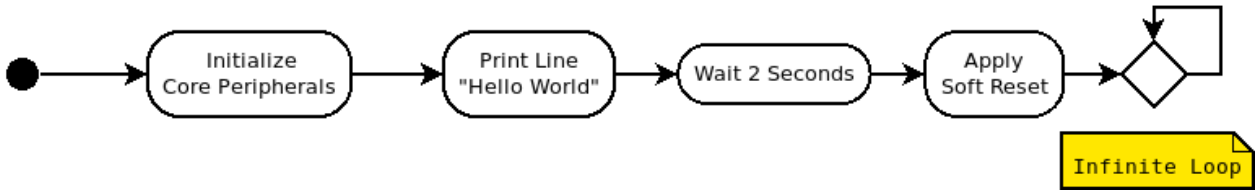

Figure 5. Hello World example activity diagram

Opening USB to Serial port on PC causes hardware reset on OtomaDUINO.

4.2 Uart Ping-Pong

This examples utilizes the usb serial I/O feature. Use the same serial port parameters to open the terminal connection. After opening serial terminal connection, type anything; OtomaDUINO will send the typed characters back to PC.

Software Description:

The software initializes the core peripherals and waits for byte from serial input. The sends it back over serial output. If there is no bytes in the input, the software waits for 100ms.

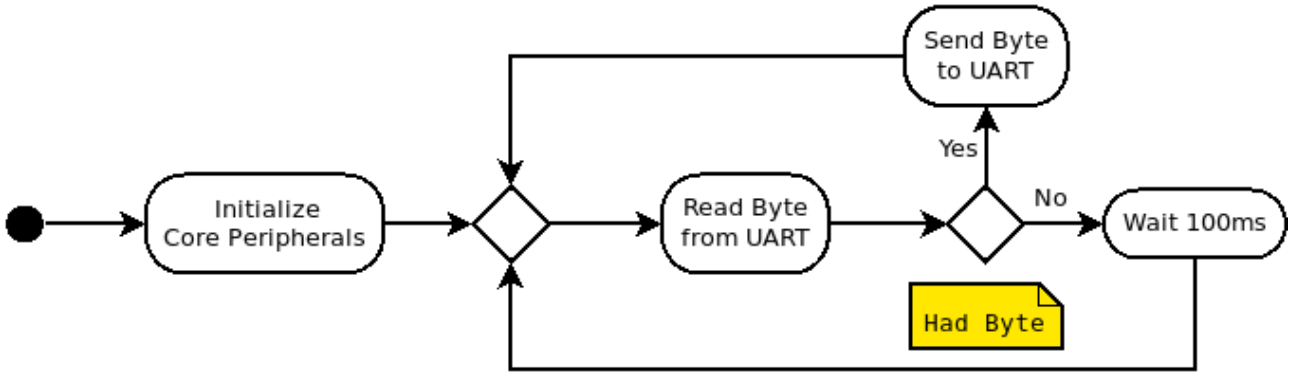

Figure 6. Uart ping-pong example activity diagram

4.3 Digital I/O and multi-tasking

This example demonstrate a simple multi-tasking scheme which can be used in many applications. In this example, MCU checks one switch and one push button; and controls a blue signal light (LED). In this example red signal light is used for power on indicator and green signal light is used for switch on indicator.

If the switch is on, the blue signal light starts to blink. And when the push button is pressed it stops blinking and it is turned on steadily. To achieve this, three different tasks were implemented these are used for checking switch state, checking button state and blinking signal light.

Hardware Description:

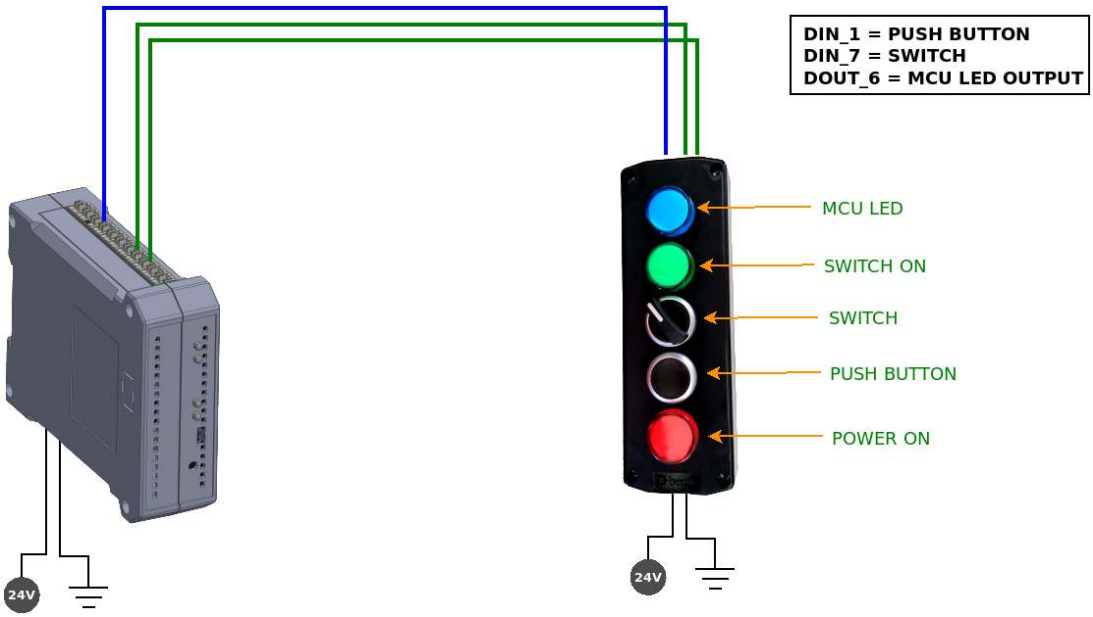

Figure 7. Digital I/O and multi-tasking example wiring

Connect the push button to Digital Input 1, the switch to Digital Input 7 and signal light to Digital Output 6.

Software Description:

In order to achieve multi-tasking we have to use a timing scheme. In this example a timing scheme was implemented by using OtomaDUINO's internal timer and a set of timestamp for each task (switch check task, button check task and led blink task).

Each of the task timestamps was check against its wait time limit. If the task waited more than the limit time, then the specific code block is executed. This type of multi-tasking is applicable only if any of the tasks do not block the execution.

At the start, the software resets timestamps, initializes core peripherals, initializes digital I/O and enables digital output (for driving signal light). Then enters infinite loop to check the time limits of each task.

If the timestamp difference is more than the limit time, the task is executed.

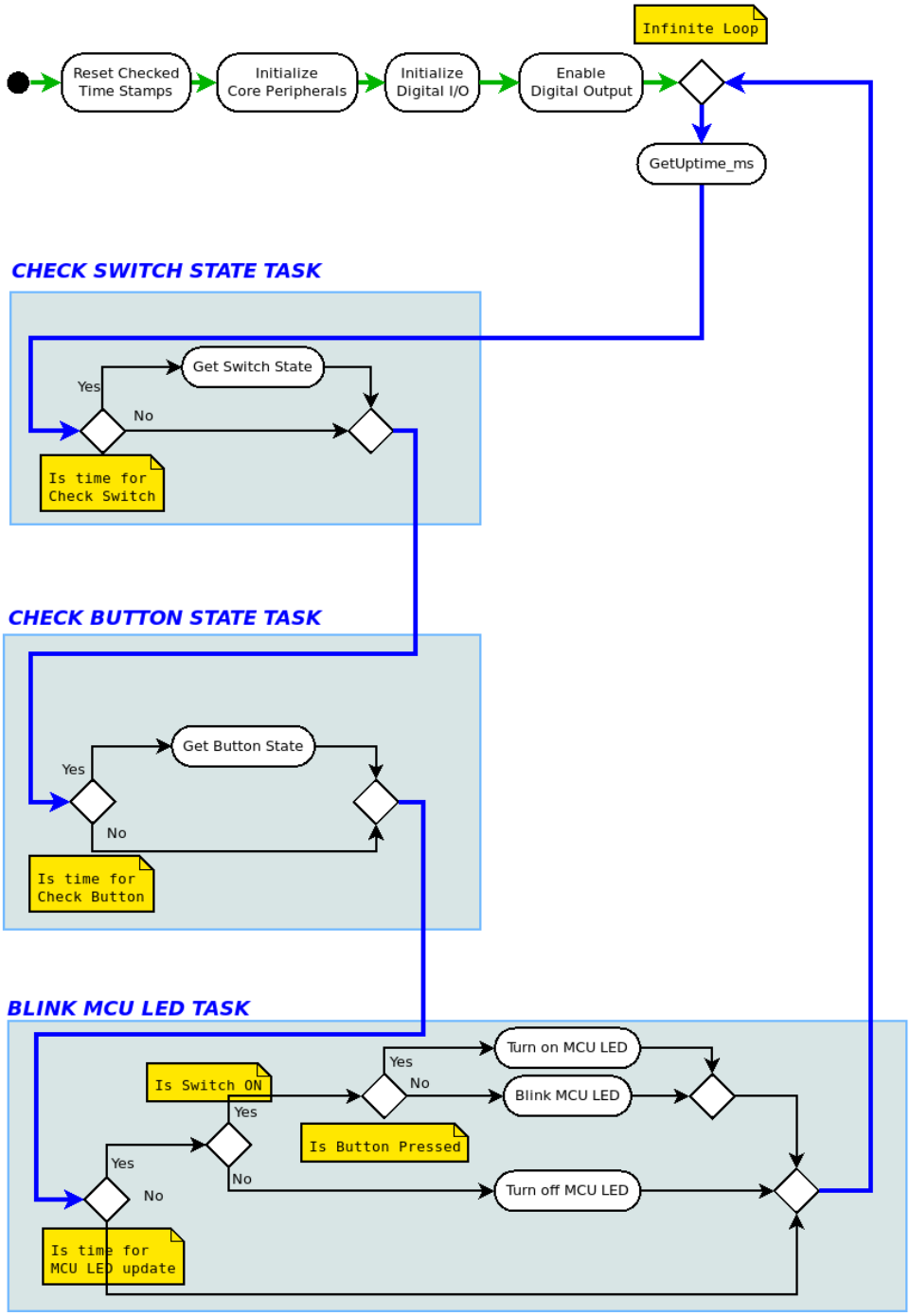

Figure 8. Digital I/O and multi-tasking example activity diagram

4.4 Driving step motor with changing speed

This example demonstrates pulse output capabilities of OtomaDUINO. It uses a state machine to apply variety of pulse outputs.

Hardware Description:

This example uses TB6600 step motor driver and a generic step motor. The setup was powered by 24VDC power supply. So, 2.2KΩ 1/4W resistor should be put between OtomaDUINO outputs and TB6600 inputs.

Connect Enable of TB6600 to Digital Output 1, Direction of TB6600 to Digital Output 2 and Pulse of TB6600 to Pulse Output 2.

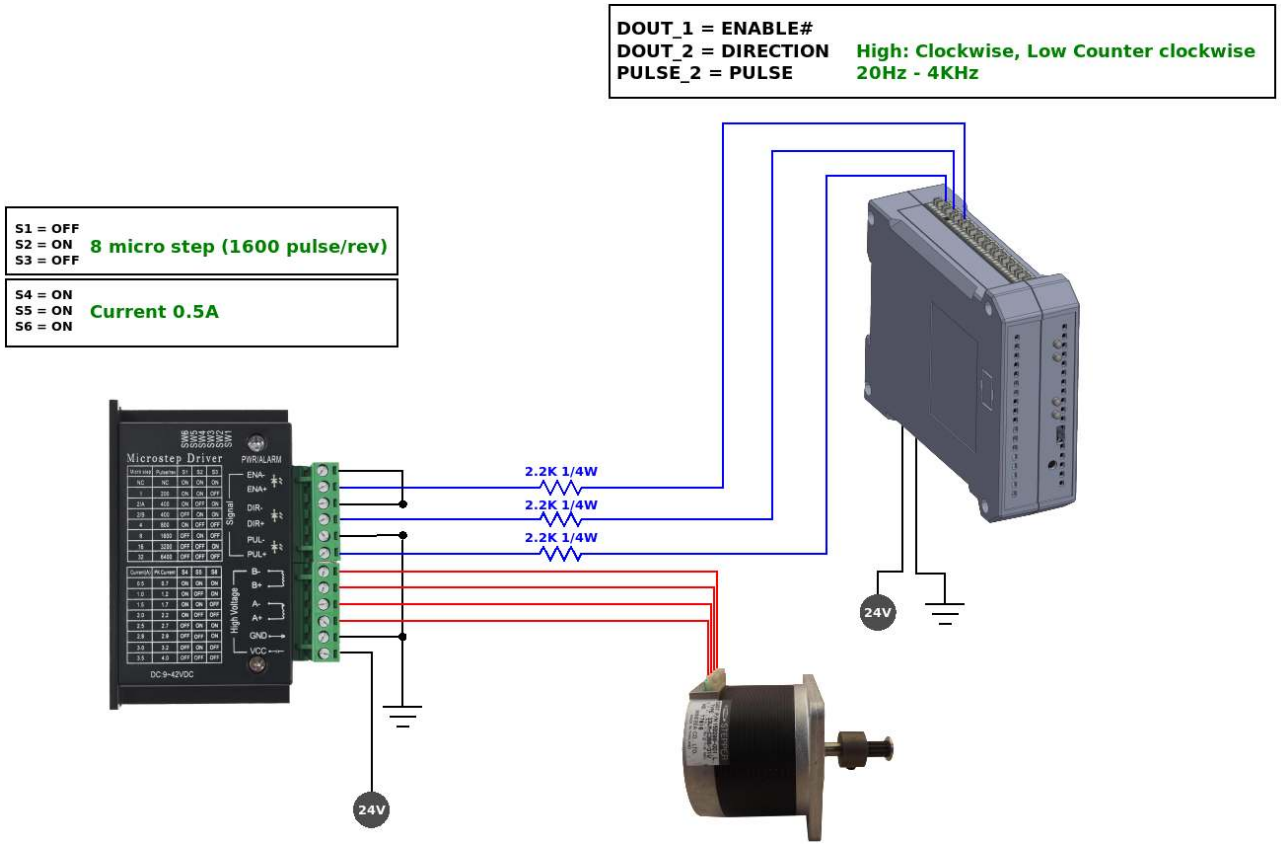

Figure 9. Driving step motor with changing speed example wiring

Software Description:

The software uses the basic multi-tasking and pulse limit count of OtomaDUINO's pulse output. In order to demonstrate different output options, the software uses a state machine with the following states: `Stop, Slow Forward, Fast Forward, Slow Down Forward, Stop and Change Direction, Slow Backward, Fast Backward, Slow Down Backward`

After the initialization, the software checks whether the pulse output is active. If not, it applies the pulse output according to the state machine state and moves the state machine to next state.

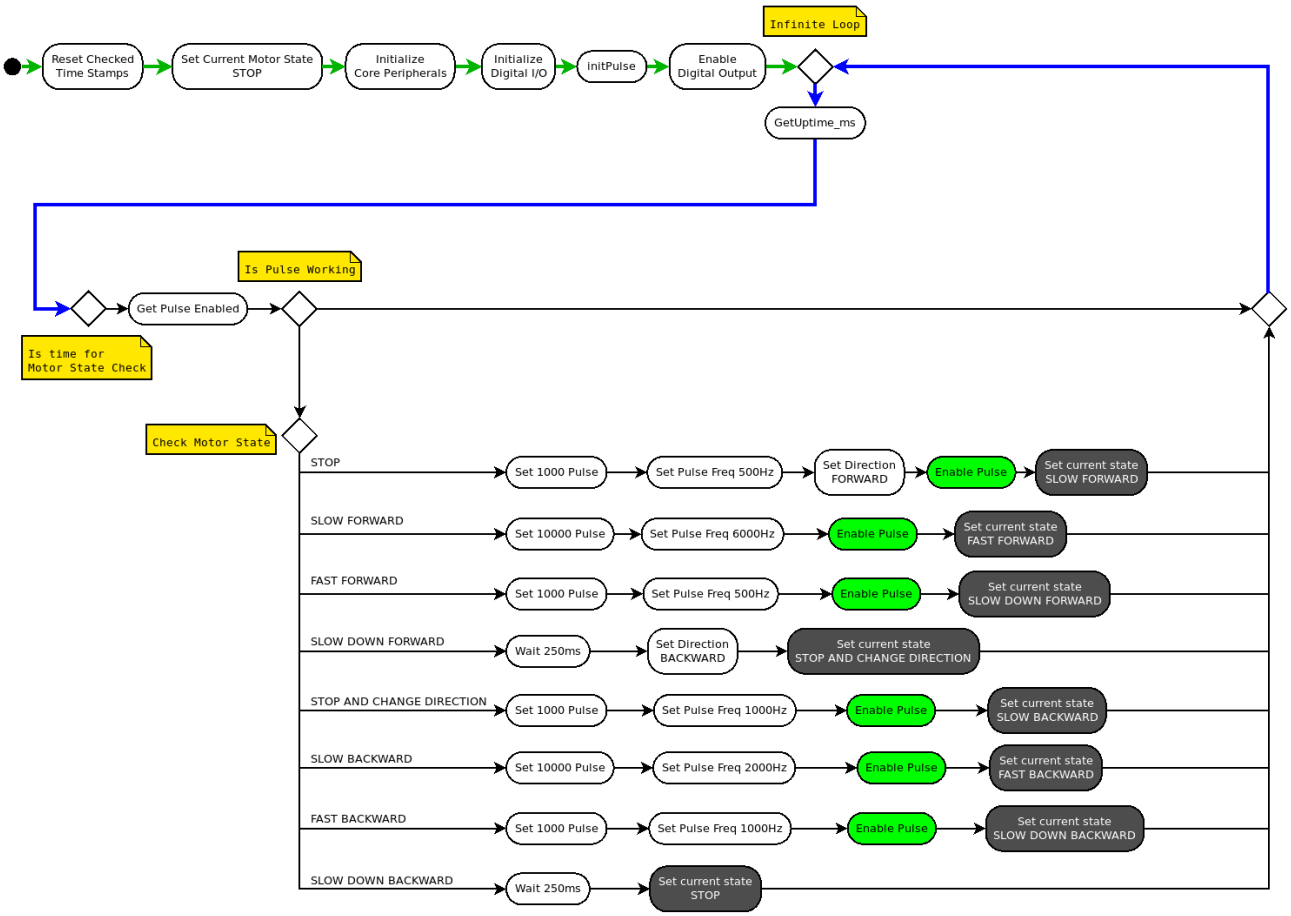

Figure 10. Driving step motor with changing speed example activity diagram

4.5 Reading differential load cell with voltage output

This example demonstrates the low voltage analog input capabilities of OtomaDUINO. OtomaDUINO has various amplifiers. One of the amplifier can be programmed. This example sets the internal programmable amplifier to maximum level (×128) and reads the differential analog output of a 10Kg load cell. The read values are send in text format to serial port.

Hardware Description:

Connect the positive output of load cell to Analog Input 1 and negative output of load cell to Analog Input 2.

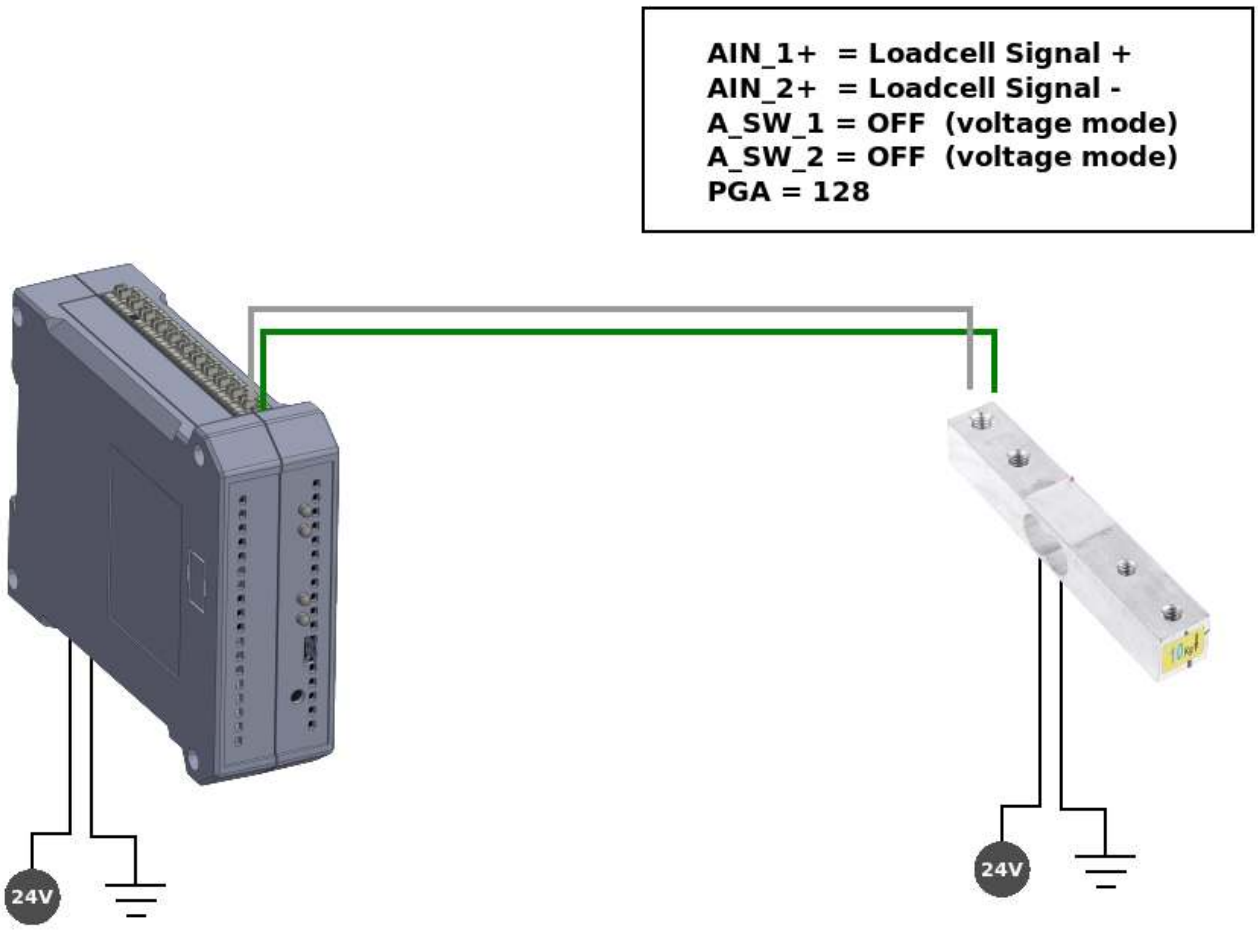

Figure 11. Reading Differential load cell with voltage output example wiring

Analog Input Setup DIP Switch Position	Switch Description
A1: OFF - A2: OFF	A1&A2 channels are used for differential voltage measurement

Table 1. Analog Input Setup DIP Switch position for load cell example

Software Description:

The software uses the basic multi-tasking and waits for analog data to be ready. If analog data is ready, it reads the data and sends the data in text format over serial port.

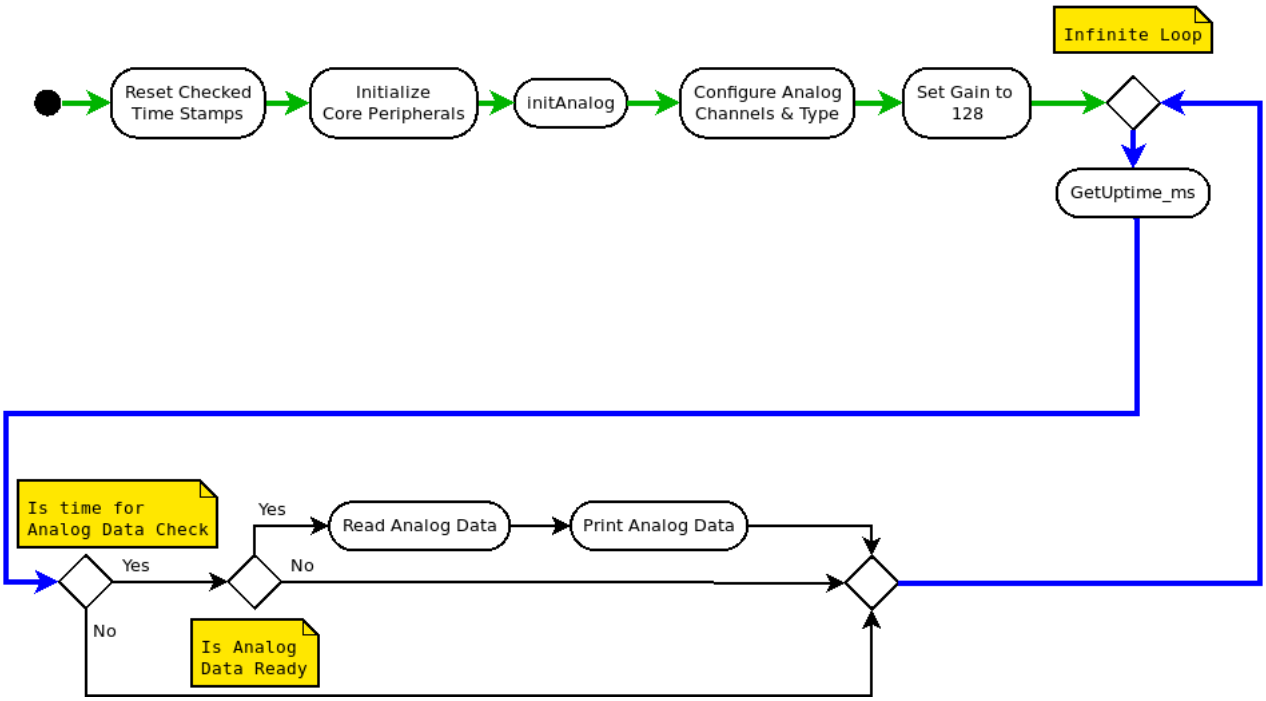

Figure 12. Reading Differential load cell with voltage output example activity diagram

4.6 Reading proximity sensor with current output

This example demonstrates the analog current measurement capabilities of OtomaDUINO. The example uses a generic proximity sensor with 4-20mA current output. The read values are send in text format to serial port.

Hardware Description:

Connect the output of proximity sensor to Analog Input 1 (A1▲) and connect ground to Analog Input 1 Current Output (A1▼).

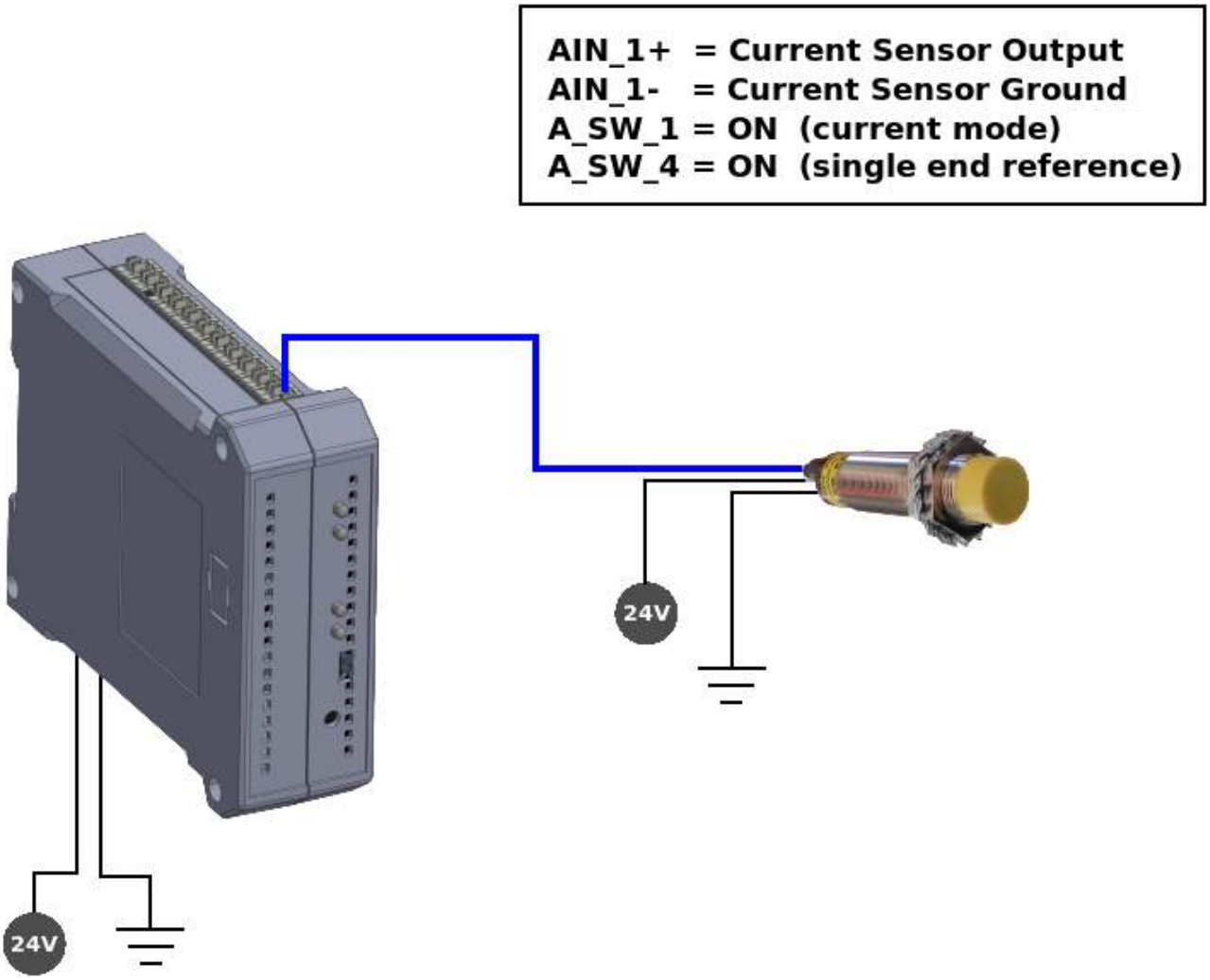

Figure 13. Reading proximity sensor with current output example wiring

Analog Input Setup DIP Switch Position	Switch Description
A1: ON - A4: ON	A1 channel is used for single ended voltage measurement A4 channel is used for single ended ground reference.

Table 2. Analog Input Setup DIP Switch position for proximity sensor example

Software Description:

The software uses the basic multi-tasking and waits for analog data to be ready. If analog data is ready, it reads the data and sends the data in text format over serial port.

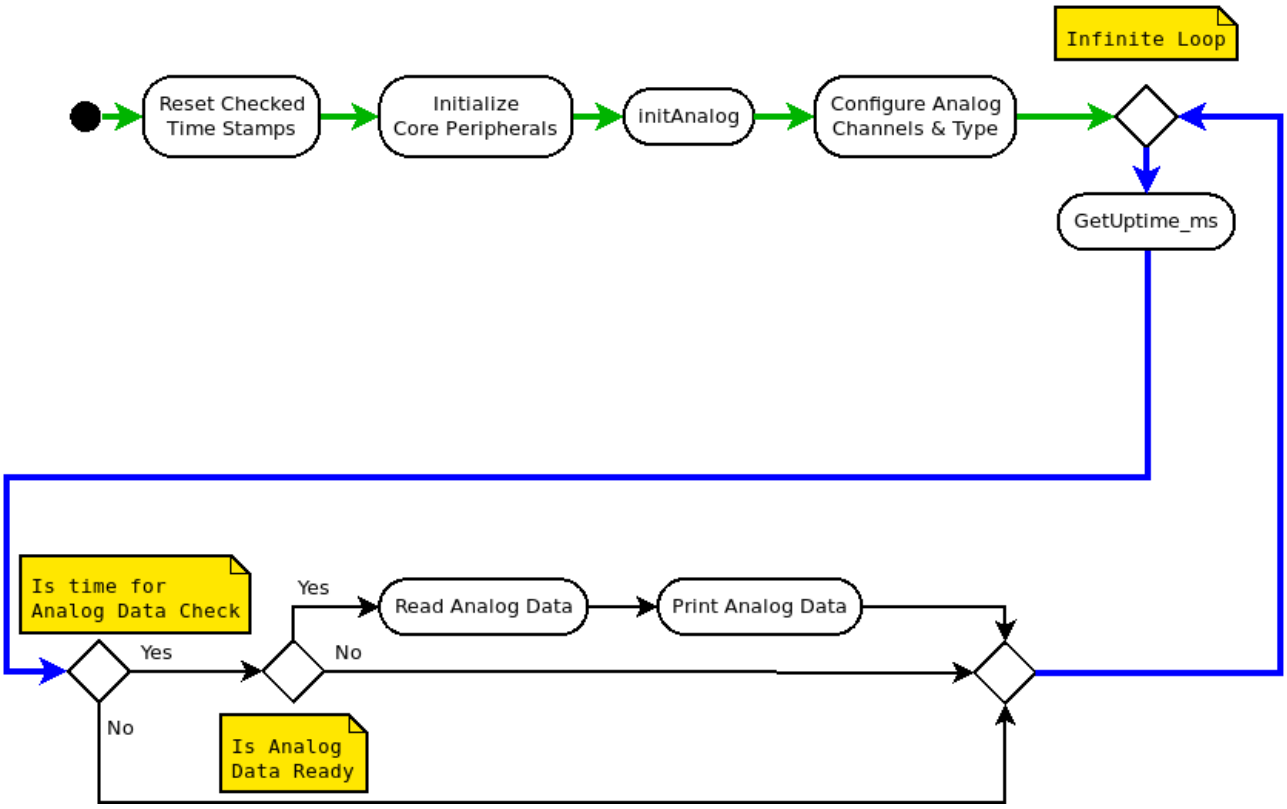

Figure 14. Reading proximity sensor with current output example activity diagram

4.7 Reading proximity sensor current output and differential Load Cell voltage output

This example demonstrates the analog input switching capabilities of OtomaDUINO. The example uses a generic load cell and proximity sensor with 4-20mA current output. The analog input is changed from load cell to proximity sensor one by one and the read values are send in text format to serial port.

Hardware Description:

Connect the positive output of load cell to Analog Input 1 and negative output of load cell to Analog Input 2.

Connect the output of proximity sensor to Analog Input 3 (A3▲) and connect ground to Analog Input 3 Current Output (A3▼).

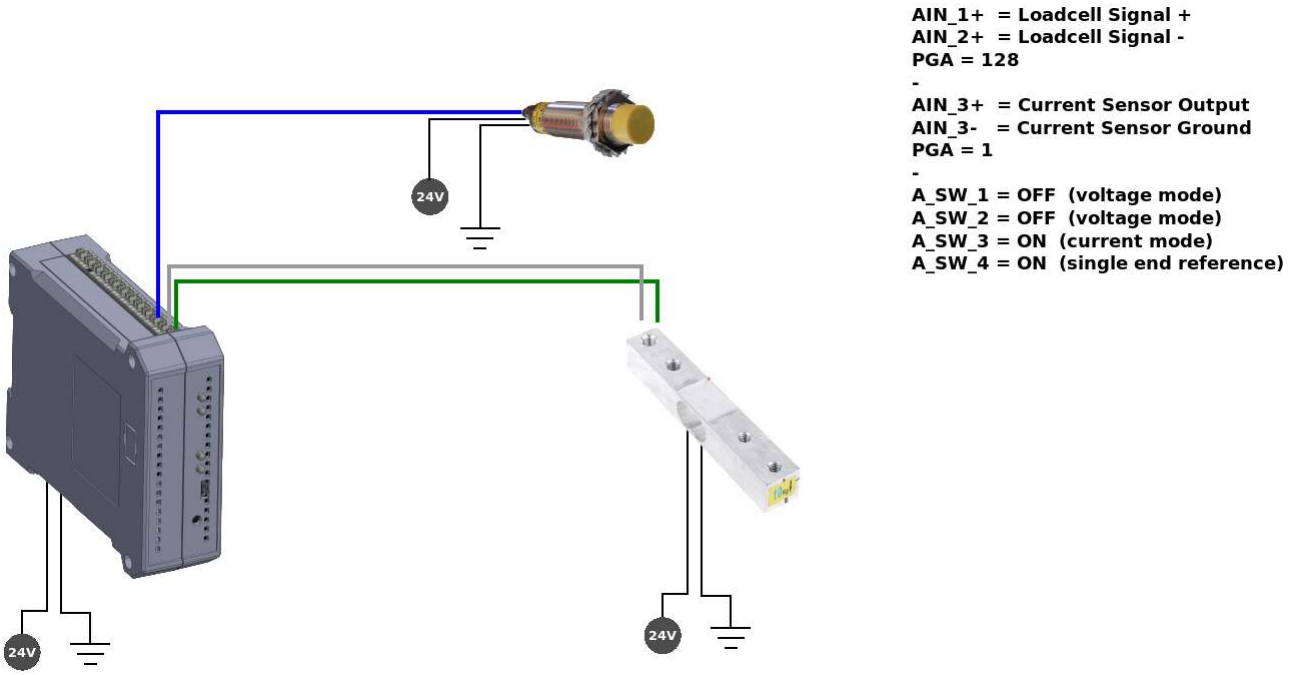

Figure 15. Reading proximity sensor current output and differential Load Cell voltage output example wiring

Analog Input Setup DIP Switch Position	Switch Description
A1: OFF - A2: OFF	A1&A2 channels are used for differential voltage measurement
A3: ON - A4: ON	A3 channel is used for single ended voltage measurement A4 channel is used for single ended ground reference.

Table 3. Analog Input Setup DIP Switch position for proximity sensor and load cell example

Software Description:

The software uses the basic multi-tasking and waits for analog data to be ready. If analog data is ready, it reads the analog data, changes the analog channel and sends the data in text format over serial port.

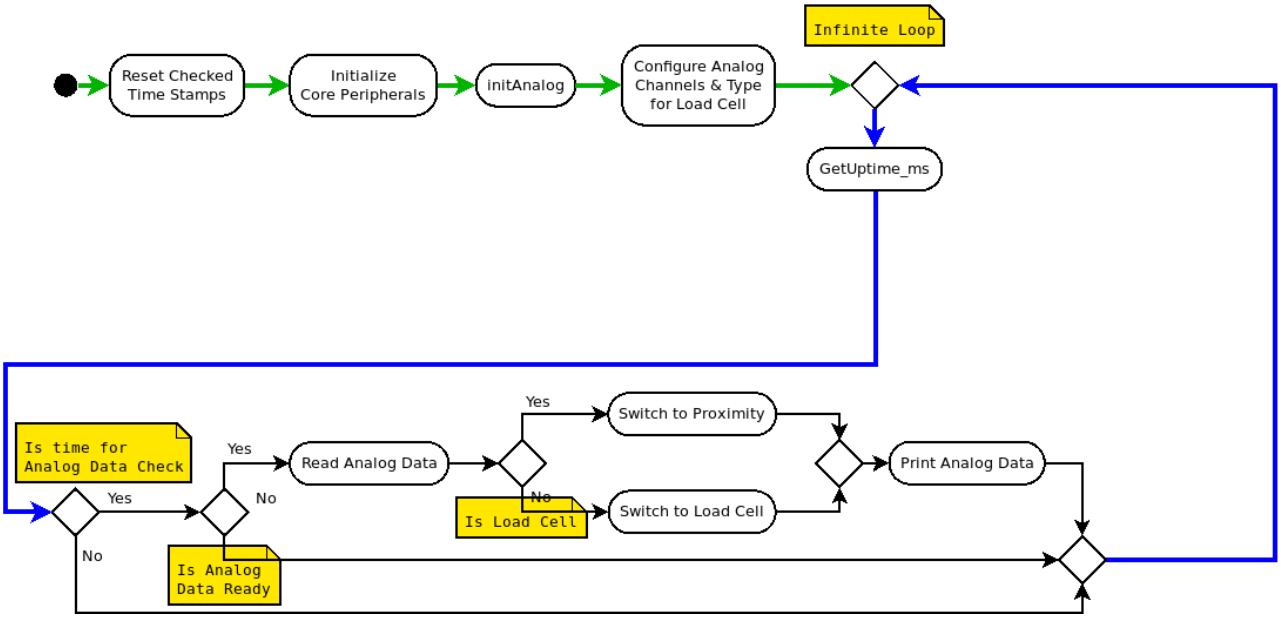

Figure 16. Reading proximity sensor current output and differential Load Cell voltage output example activity diagram