Pico Radar Speed Sign Controller

# Features

* Pi Pico Microcontroller (RP2040: 32-bit Dual Core ARM Cortex M0 +)
* 16x 5V Logic Outputs
* 9-16VDC Input Range
* 5V 3A DC-DC Switching Converter with Extra +5V Outputs
* Dedicated input for HB100X or similar pre-amplified doppler radar sensor (requires PWM input).

# Applications

* DIY Radar Speed Sign
* Two Digit Seven-Segment Display
* Other stuff that requires a power supply, a microcontroller, and a bunch of 5V digital outputs.

# Principle of Operation

There’s not a lot on this guy! Just a switching power supply, a microcontroller, and some shift registers. This makes wiring up a dedicated speed sign a bit easier, as it combines a few elements of the circuit onto the same PCB.

The onboard Pi Pico reads a PWM waveform from a pre-amplified doppler radar sensor via the HB100X port. The timing between rising edges is used to determine a doppler shift between radar signals transmitted and received from the radar sensor, which corresponds to the velocity of a moving target.

Speed information can be displayed on a seven-segment display using two shift registers that are connected to the Pi Pico. These shift registers are controlled via SPI and also include an output enable pin that allows the digits to be dimmed via PWM. Shift register outputs are 5V TTL, allowing control of a wide range of peripherals.

# Quick Stats

|  |  |  |
| --- | --- | --- |
| Parameter | Value | Note |
| Input Voltage | 9-16VDC | Reverse polarity protected. We gotchu. |
| Logic Output Voltage | 5V |  |
| Logic Output Current | ±6mA | Total output current ±35mA for each shift register (e.g. outputs 2A-2H). |
| Aux 5V Output Current | 3A\* | Measured with no load from the rest of the board. |

# Software Setup

The RP2040 microcontroller in the Pi Pico used on the Pico Radar Speed Sign Control Board runs MicroPython. Code can be re-flashed via a USB connection to the Pi Pico. Python instructions can also be run via the Pico’s Read-Execute-Print-Loop (REPL) which is available over the serial connection.

Use of a Integrated Development Environment (IDE) like Thonny or Visual Studio Code with the Pico-W-Go extension is recommended for flashing or writing new code.

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Author** | **Note** |
| 0.1.0 | 2023-07-14 | J. McNelly | Alpha release. |