Astoria HS Riverbots FRC #6845

Astoria High School, Astoria Oregon USA
https://riverbots.org

The Stupid Ridiculously-Cheap Drive Encoder (“StuRCDE”)

By Victor Croddy

Introduction

“Necessity is the mother of invention” is a proverb that can be traced back as far as Aesop’s fables. It should be no surprise that this remains true in FRC robotics, especially for teams that may have less resources than ideal and need to innovate or compensate for that. Fundamentally, our Stupidly Ridiculously-Cheap Drive Encoder™ embodies this principle by being a functional replacement for a true drive encoder at a fraction of the cost. Here, Innovation in Control is exemplified by innovating around the financial strain that other digital encoders would place on a robotics team.

Design Summary

Our cheap drive encoder is able to, for the purposes of driving during Autonomous mode, track where our robot is on the field at any given time. With a circuit installed onto what is essentially a car stereo’s volume knob (that is installed into the outer facing bearing of our drive wheels), we can count the number of rotations our robot’s wheels make and thus calculate how far it has traveled from its starting point. Some other parts include:

- A simple, custom-printed circuit board
- 4 pull-down resistors (can theoretically range from 100 ohms to 100K ohms; here, 1 kilo ohm resistors are used)
- 4 three-pin connectors for connection to the DIO ports of the RoboRio

The DIO is hooked up to the positive side of a pull-down resistor, which reads 0V most of the time with +V spikes as the encoder turns.

Design Requirements and Goals

Some requirements for our solution included that it must fit between the frame’s flanges as well as between the frame and the bumper; it must count the rotations the wheel makes; it must be significantly cheaper than standard drive encoders; and relatively easy to produce.

The requirements for this part were more or less that of any other encoder. It needed to count the number of rotations a wheel would make in a form that is translatable to a RoboRio, fit within the perimeter of the robot itself, must be able to physically connect with the drive wheels or gearbox, and be available from what materials were immediately available.

The goals for this part were for the part to be cheap to produce, relatively easy to make, and reusable. Depending on the resources available, the cost factor of our goals could be considered closer to a requirement, and so could the other two as a result of time constraints.

Physical Interface

The encoder's shaft was fit snugly into the outside bearing of the drive wheel. To ensure that it rotates with the wheel, a small spacer of foam has been stuck onto the shaft (the part of the encoder that is supposed to turn). Two brackets are also mounted onto the base of the encoder shaft and secured by a nut securely tightened onto it. Both extend out on both sides of the encoder and are bolted to the frame of the robot (some bending of these aluminum brackets was required).

The bracket closest to the frame of the robot is to secure the encoder in place, but the outermost encoder has slots cut into them to prevent the encoder itself from rotating with the wheel. Otherwise, it may turn and twist the wires and break the connection.

Noticeably, this attaches to the *outside* the aluminum frame of the robot, rather than on the *inside* via the motors' gearbox. An additional advantage of these encoders is that they are reusable and are not destroyed upon removal.

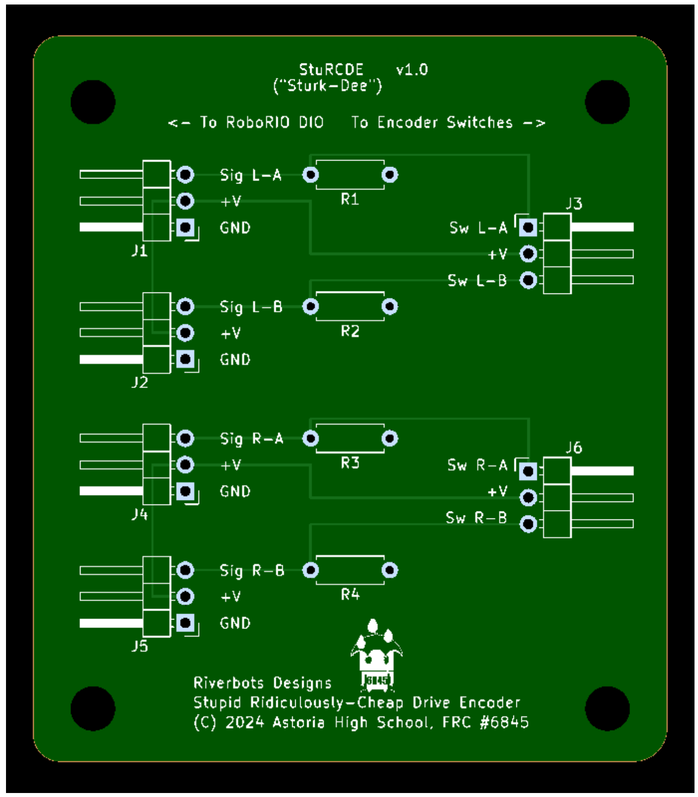
Electrical Interface

The electronics of the StuRCDE are relatively simple and do the job of a standard encoder, with some assembly. Inside the encoder is a mechanical switch that, when turned, closes a circuit to send a pulse of electricity at a specific length such that, when turned one direction, it sends an electrical signal to one DIO port closely-followed by the other DIO port (and reversed when running in the opposite direction).

Astoria HS Riverbots FRC #6845

Astoria High School, Astoria Oregon USA
https://riverbots.org

The electrical circuit added to this encoder consists of two pull-down resistors for the purpose of preventing a short within the encoder (bad!). Theoretically, as long as these resistors are not miniscule or gigantic in ohm value, it shouldn't matter what they are, as long as they provide some resistance to the 5V input. Probing one side of the encoder and a point before the resistor should read 0 ohms, while at the same side of the encoder and another point after the resistor should read 1K ohms relative to ground.



Code Interface

The step after electrically configuring the encoder is interpreting the signals it actually sends to the RoboRio. This is a simple process: in the main java file, the encoder adds to a value X when moving forwards, and subtracts when moving backwards. The point where the electrical interface meets the code interface is at 4 DIO ports. One encoder occupies two: one trigger sequence when "forwards" is read, and the other when "backwards" is read.

This is electrically reasonably similar to the US Digital E4T without the need for active components and uses the same WPILib Encoder interface.

Analysis

The single greatest advantage of this part over the AndyMark E4T is the monetary cost. Two AndyMark E4T encoders (excluding shipping costs) are in total $86, as of 3/5/2024. The total cost for the PCB and encoders bought for this project would, as of 3/5/2024, barely exceed $26 (excluding shipping costs). Not to mention, this is the cost to buy 10 of each - the unit cost of each PCB and encoder, then, can be estimated to be $2.60. While the E4T is rated for a much longer duty cycle before wearing out at greater precision, it doesn't really matter when the encoders only need to last for a few dozen operational hours over the course of an FRC season and sub-mm precision is unnecessary Additionally, the E4T parts are essentially disposable/non-reusable as a practical matter once removed from a gearbox anyway, so avoiding this recurring year-over-year cost is a solid tradeoff.

Conclusion

Autonomous requires a way to electronically read how much the robot has moved via wheel revolutions. The way to do this is with an encoder, but that posed a high financial cost to the AHS Riverbots team. So, we innovated upon cheaper, non-standard encoders readily available via a custom-printed circuit board that allowed us to both reuse encoders in the future and save on costs - especially important for a team with less resources such as the AHS Riverbots.

Although this PCB was used specifically for encoders attached to drive wheels, it could be utilized wherever it is able to have an encoder physically interface with a motor. Technically, one does not *need* the PCB to create the same circuit, but that requires hours more of painstaking soldering - a significant time loss for any team.

Resources

All design files (using the Kicad EDA software) are at:
https://github.com/riverbots42/2024/tree/main/encoder
Boards were produced at PCBWay using the native Kicad export plugin to PCBWay.

The rotary encoders used were UXCell 360-Degree Rotary Encoders sourced in bulk from Amazon.