

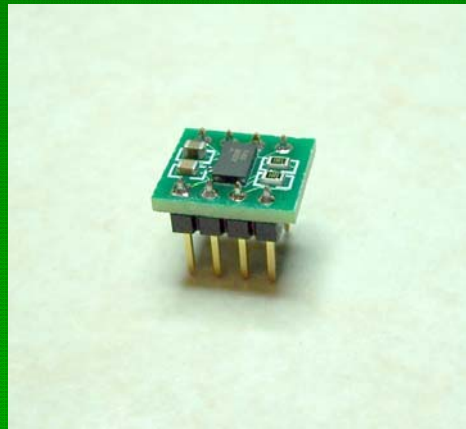
3AD Usage Guide

Raymond Allen
Embedded Systems Architect
Rayslogic.com, LLC

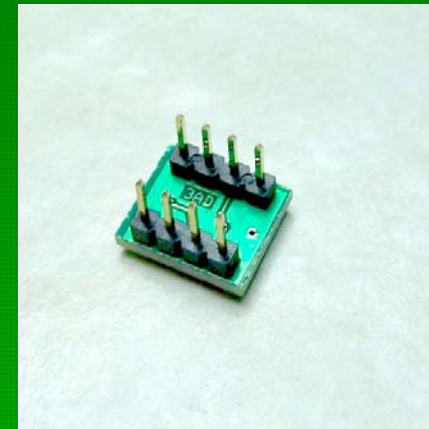
Rev.1 26APR10

The 3AD is a low cost 3-axis digital accelerometer in a 8-pin DIP package

Top View:



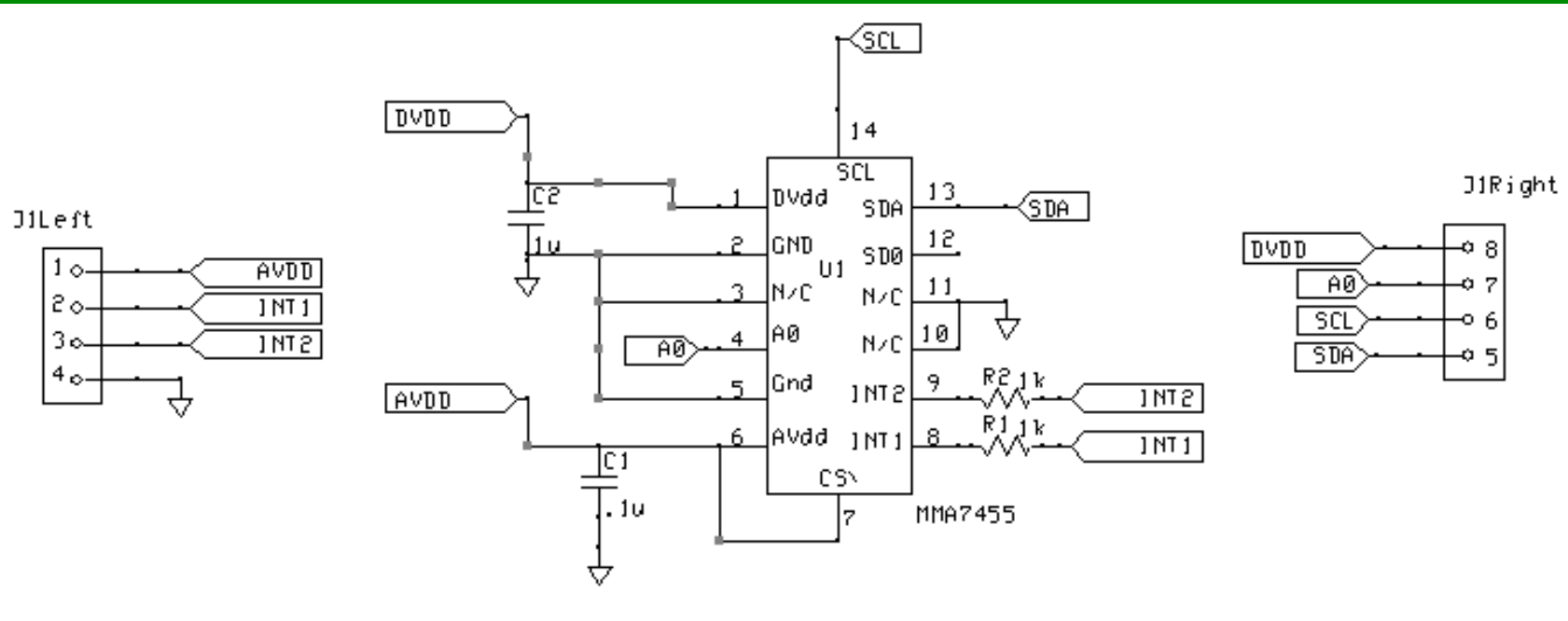
Bottom View:



■ Key Features

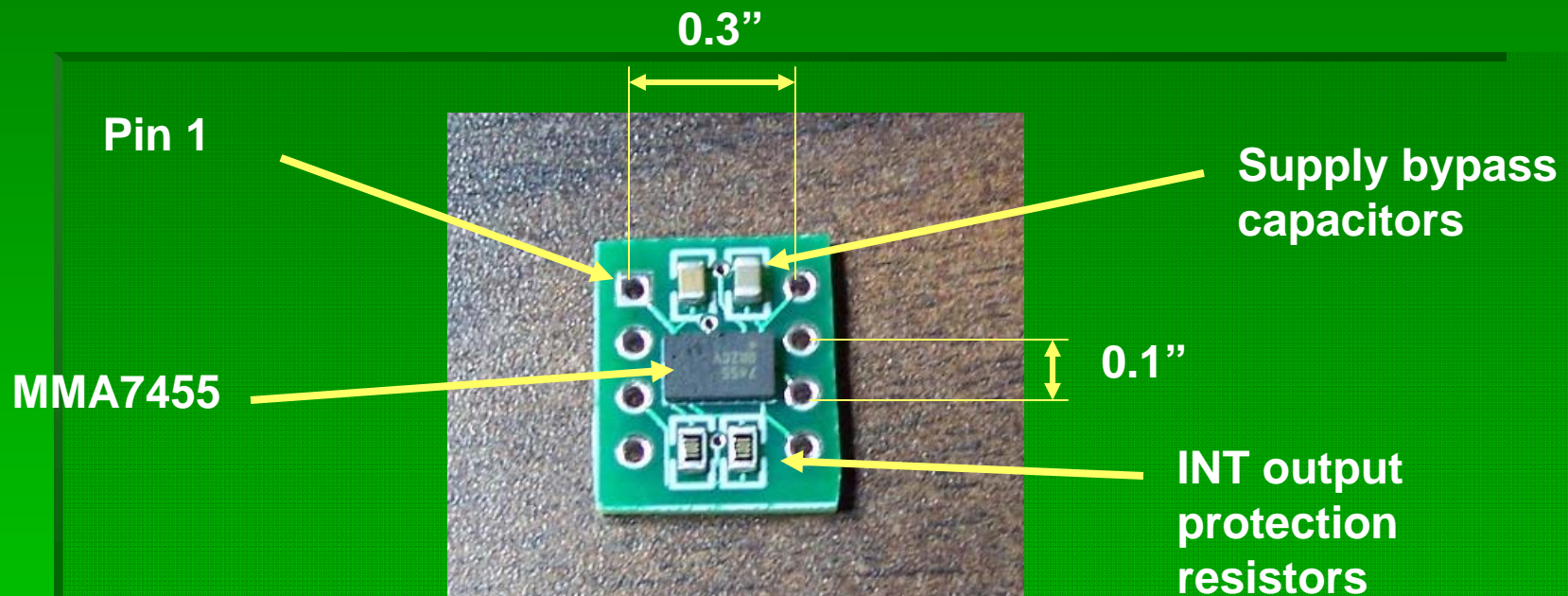
- Based on the Freescale™ MMA7455 accelerometer chip
- I2C digital interface
- 8-Pin DIP package, 300-mil width
- Compatible with the extra EEPROM socket of the Propeller Platform (available at GadgetGangster.com)
- Low cost (\$8 introductory price)

3AD Circuit Schematic



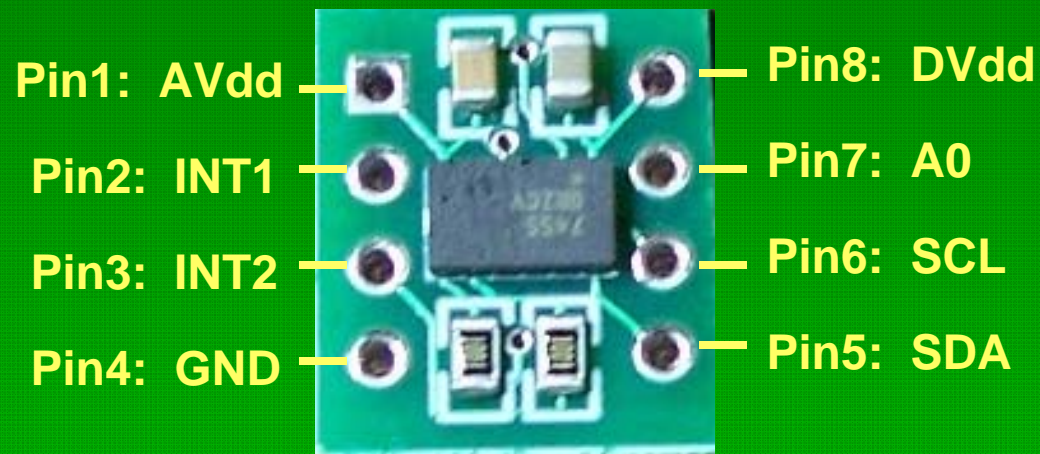
- The 3AD comprises the MMA7455 chip, bypass capacitors for analog and digital supplies and protection resistors for the interrupt outputs

3AD Layout Overview

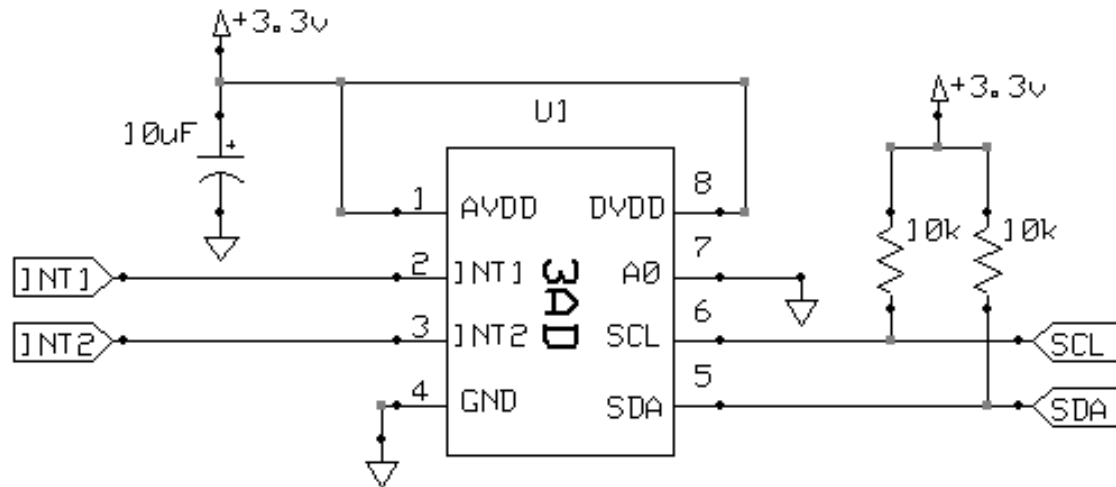


- Pin#1 can also be identified by it's square pad

3AD Pin Diagram

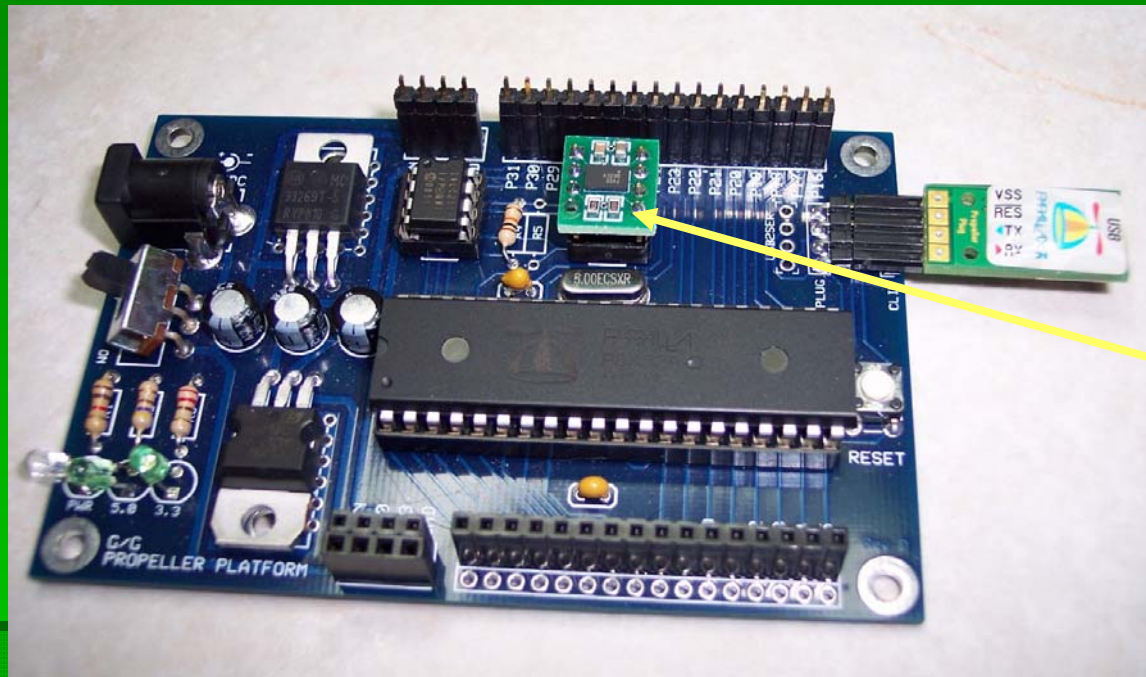


3AD Typical Usage Schematic



- An external 10µF tantalum capacitor located close to the 3AD is recommended.
- If connecting to an existing I2C bus, the 10 kΩ pull-up resistors may already be present. (The Propeller Protoboard already has the required pull-up resistor on the I2C bus of pins 28&29)
- If the two interrupt outputs are not required they can be left floating or grounded.

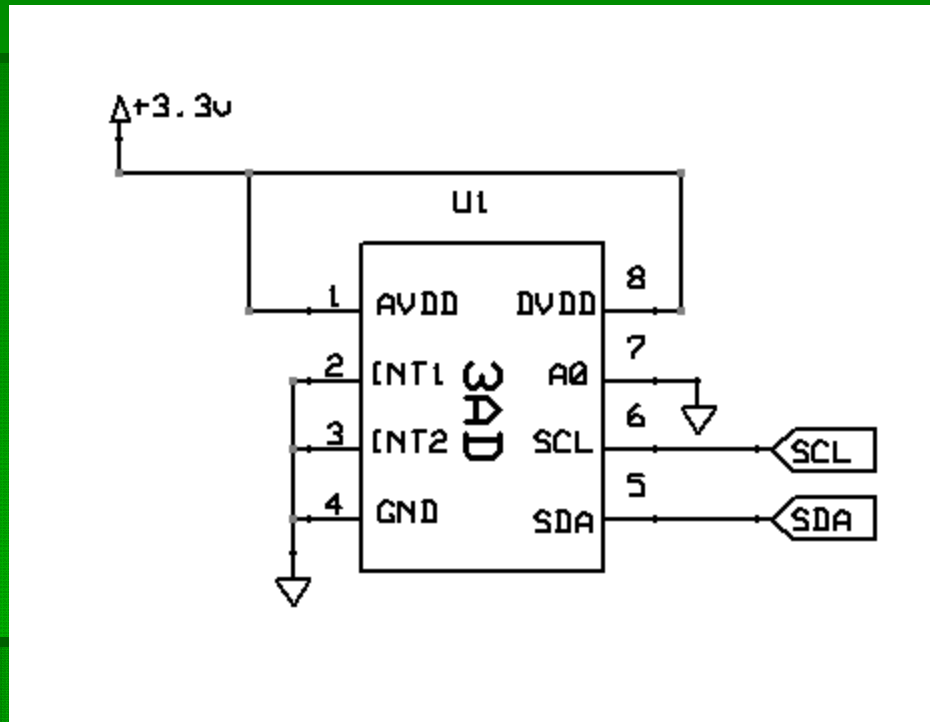
3AD is Compatible with spare EEPROM socket of the Propeller Platform



3AD

- The 3AD was specifically designed for compatibility with the spare EEPROM socket of the Propeller Platform.
- A DIP socket such as the Digikey A24794-ND allows for easy insertion and removal of the 3AD from the Propeller Platform

3AD on Propeller Platform Schematic



- This schematic shows the electrical connections when the 3AD is plugged into the Propeller Platform's spare EEPROM socket.
- The interrupt pins are connected to ground (and therefore not useable), but the onboard 1k resistors prevent any problems.

Example I2C Driver Provided

- The MMA7455 is also a component of the Rayslogic.com PSB and the PSB's I2C driver "PSB_I2CDriver" is provided for communicating with the MMA7455.
- The available "3AD_Test" project shows how to use the I2C driver to get X,Y&Z data from the 3AD.
- Before use, the 3AD must be initialized with this command to the I2C driver:
 - `i2c.WriteAccelReg($16,%0101)`
- Then, you can get accelerometer axis data with calls to the I2C driver like this:
 - `x:=i2c.GetAccelX8`