Building Robots That Can See

Scott Garman
Intel Open Source Technology Center
scott.a.garman@intel.com

ELC Europe ● Edinburgh ● 25 Oct 2013

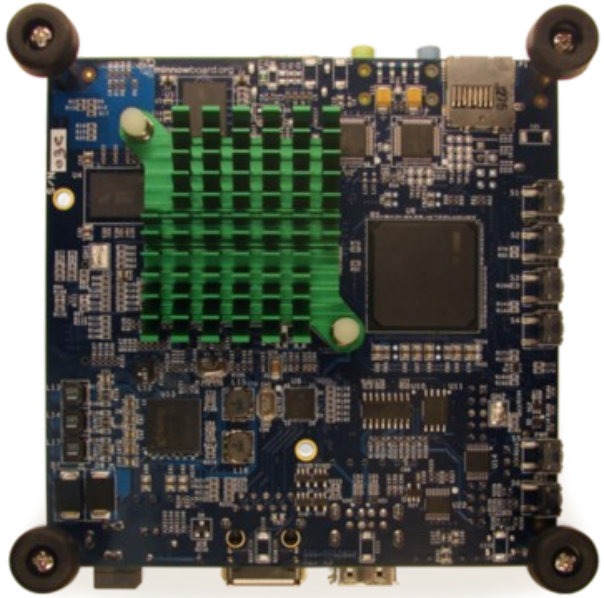
Who Am I?

- Software Engineer on the Yocto Project
- Technical Evangelist for MinnowBoard and Yocto
- Embedded Enthusiast, Bike Geek, Maker
- Sometimes obsessed with puns (OK, always)

If you think of any good fish puns, be sure to let Minnow

What is the MinnowBoard?

The MinnowBoard is an Intel Atom – based board which introduces the Intel Architecture to the small and low cost embedded market for the developer and maker community. It has exceptional performance, flexibility, openness and standards for the price.



Meet the MinnowBoard

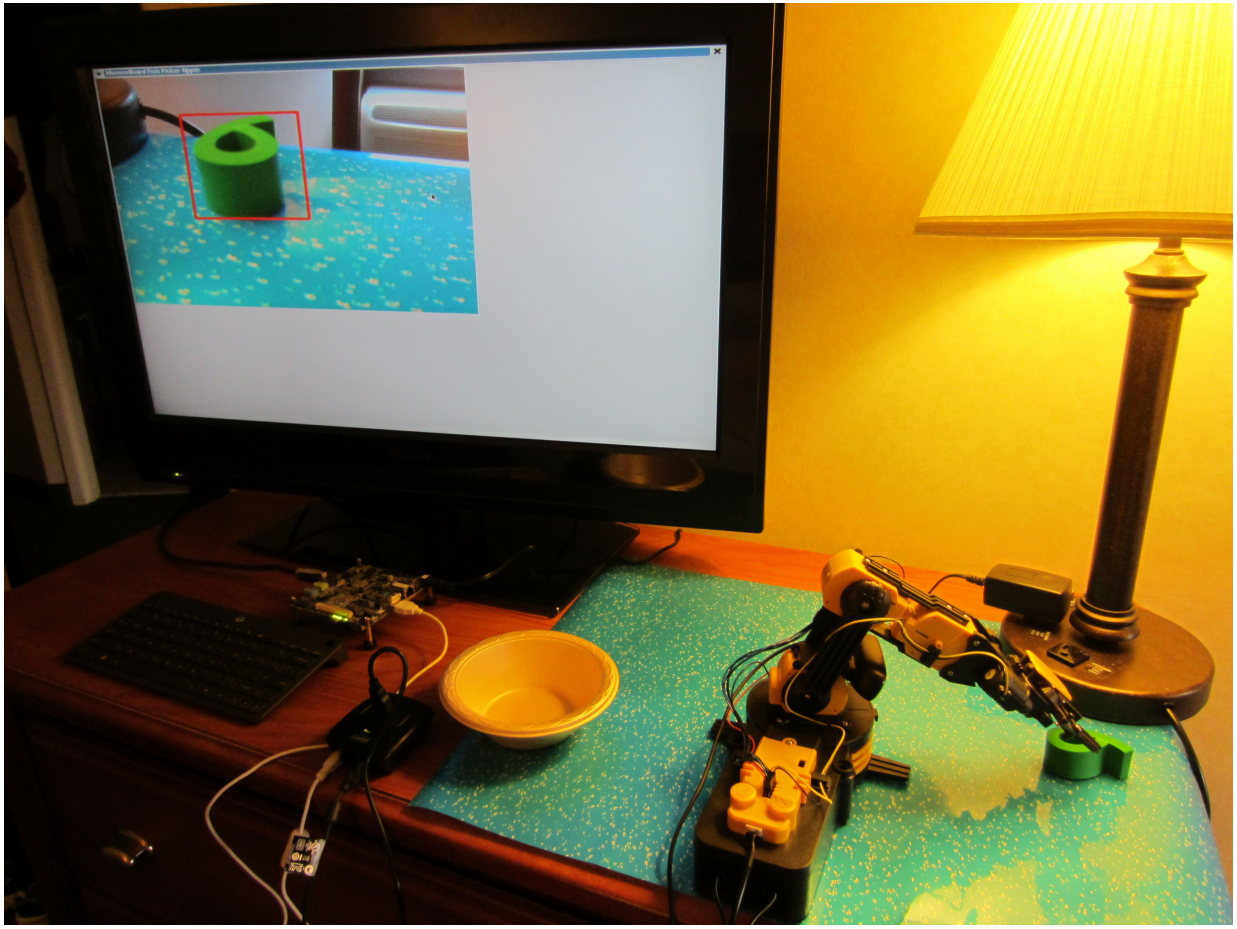
- An open hardware Atom platform
- 1 GHz CPU w/ HT and VT-x, 1 GB RAM
- Strong I/O performance powered by PCI Express:
  - SATA 3 Gbps
  - Gigabit Ethernet
- Hobbyist-friendly but scales up to higher workloads and serious embedded applications
- Embedded I/O – SPI, I2C, GPIO, CAN

Meet the MinnowBoard

- Expandability via MinnowBoard “Lures”
- 10.7cm x 10.7cm square, MSRP $189 USD
- Ships with Angstrom Linux distro, Yocto Project Compatible
- Now shipping – EU distributors: Farnell, Tigal

Learn more at www.minnowboard.org

Meet the MinnowBoard Fish Picker-Upper



Project Goals

- Develop a fun & family-friendly demo project that also relates to a serious embedded embedded application space
- Make it easily replicatable and affordable (robot arm < $100 USD)
- Make use of OpenCV to demonstrate object detection and autonomous robot control

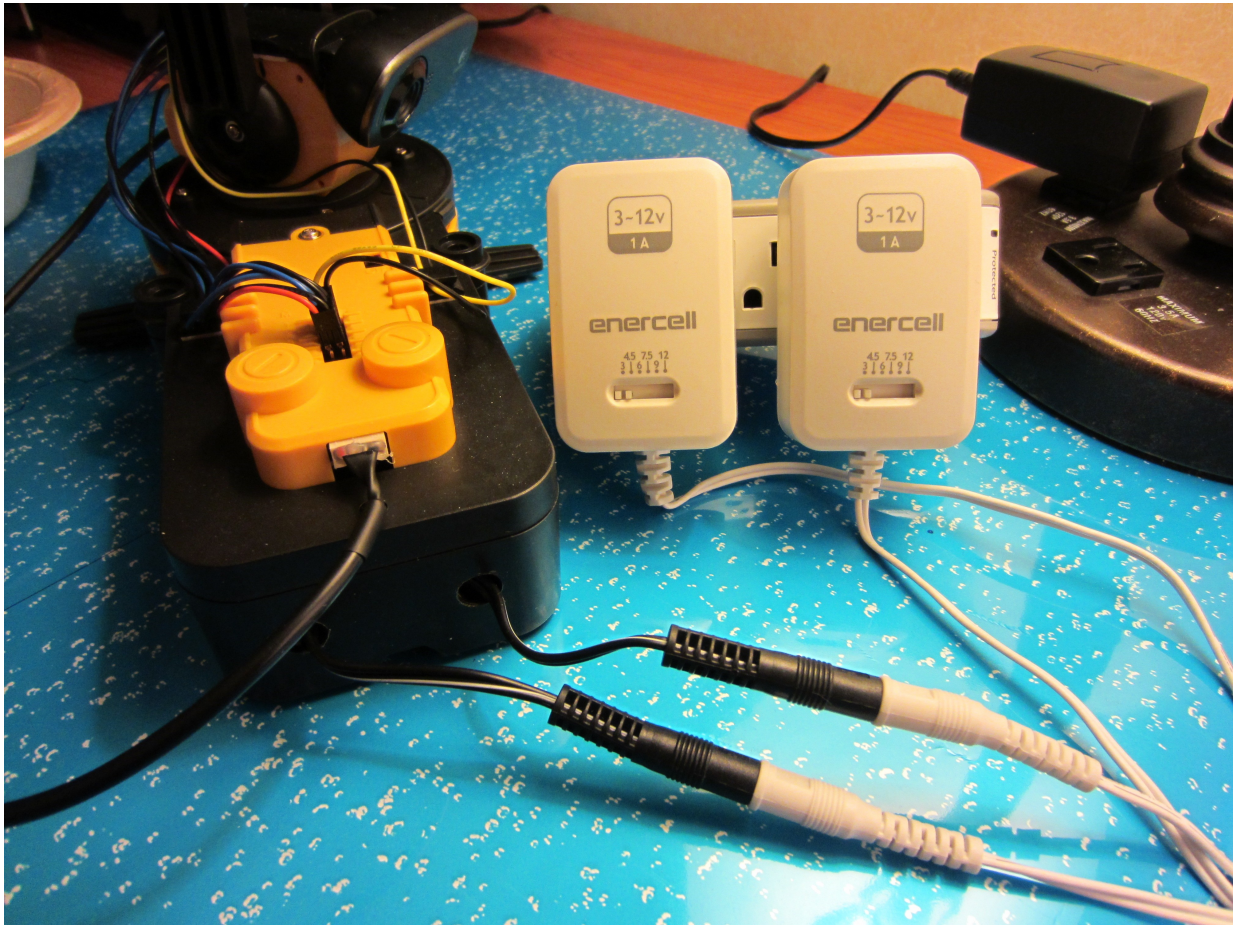
Operation

- USB webcam mounted to robot arm rotating base used to scan and identify foam “fish”
- Arm centers on and picks up desired fish
- Moves the fish to a dinner plate

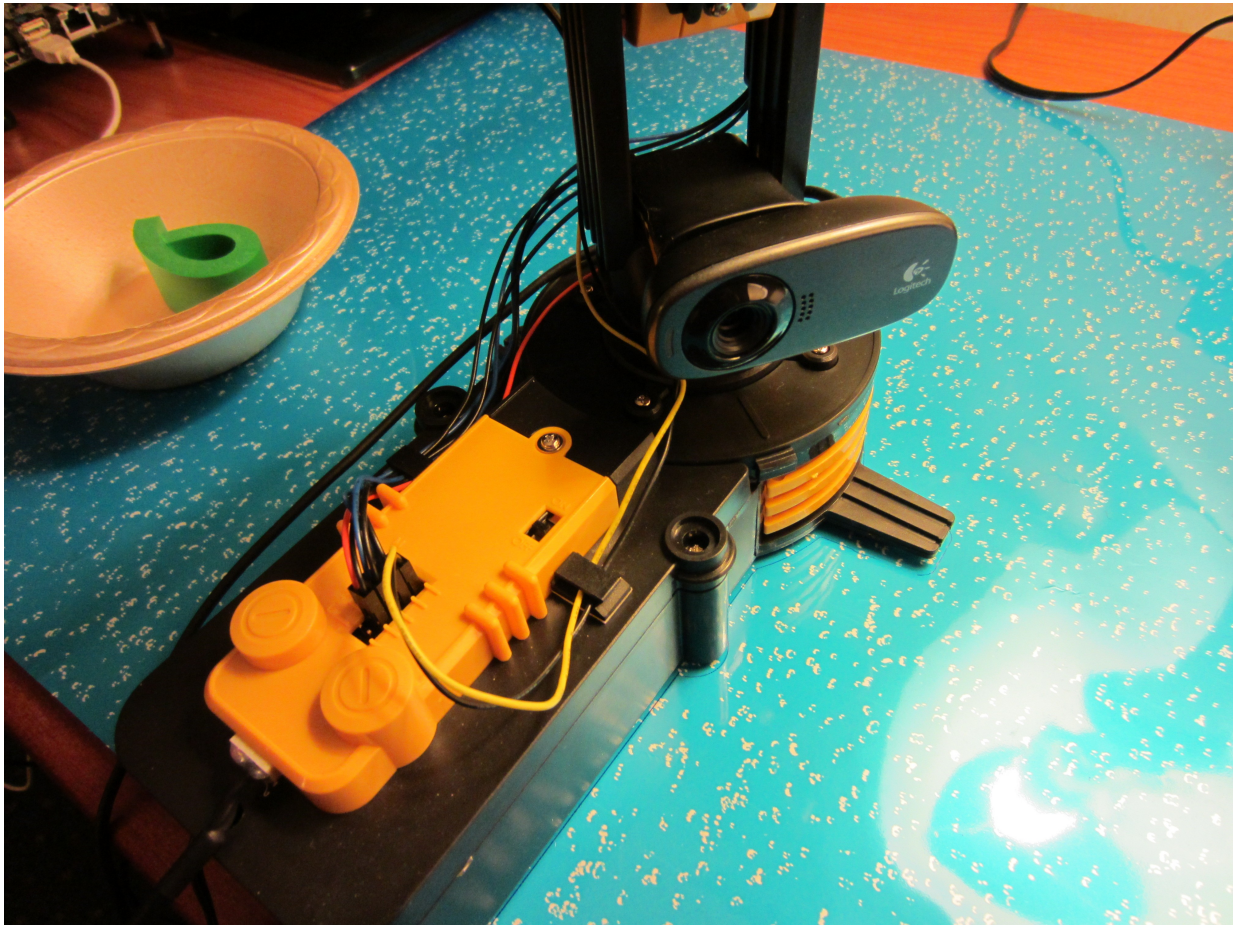
Challenges

- Webcam too heavy and not wide enough field of view to mount on the grippers
- Lack of servo motors mean lots of trial & error to figure out motion timing values
- OpenCV object detection challenging to get precision from (especially sensitive to lighting conditions)
- Embrace constraints: standardize commands to pick up and put down the fish, use only the base rotation axis for object detection

Power Supply Modification



Webcam Mounting



OpenCV Intro

- Open Source (BSD licensed) computer vision and machine learning library started by Intel in 1999
- More then 2500 algorithms in object detection and classification, motion tracking, and complex image processing
- Cross-platform, offers C, C++, Python and Java interfaces

OpenCV Object Detection

- Machine learning algorithm can be “trained” by feeding it examples of images with the object (positive samples) and ones without the object (negative samples)
- Haar Classifiers are a (relatively speaking) computationally efficient way of identifying similarities between images (positive samples)

Training the Haar Classifier

- Begin with large sets (hundreds or more) of positive and negative images
- Mark the Region of Interest (ROI) in each positive sample using a utility such as ObjectMarker – generates a text file with the image filename and ROI coordinates
- Convert the ROI text file into a training sample vector file using opencv_createsamples
- Run opencv_haartraining against the positive sample vector file and a list of the negative sample image filenames

OpenCV Demos

Live demo overviews of the following commands:

$ ObjectMarker pos_desc.txt positive_images/

$ opencv_createsamples -info pos_desc.txt -vec pos_samples.vec

$ opencv_createsamples -vec pos_samples.vec

$ opencv_haartraining -data haarclassifier -vec pos_samples.vec -bg negative_filenames.txt -npos 402 -nneg 354 -nstages 8 -mem 2000

...and then a demo of the MinnowBoard Fish Picker-Upper!

TurtleBot - Uses Microsoft Kinect for Vision

TurtleBot™

Adding Yocto Project Support

- Need to add opencv and its dependencies (python-numpy, v4l-utils, libav, swig, etc)
- The meta-openembedded repository has just about everything we need
- So, grab recipes from meta-openembedded and create a custom layer, meta-robot-opencv-demo
- Define a custom image recipe that boots to X (x11-base) and include the desired opencv packages

meta-robot-opencv-demo Layer Structure

```
meta-robot-opencv-demo/
    conf/layer.conf
    recipes-robot-opencv-demo/
        images/robot-opencv-demo-image.bb
        libav/
        opencv/
        python/(python-numpy, python-usb, etc)
        swig/
        v4l2apps/
        webm/
        wxwidgets/
        x264/
```

Image Recipe – robot-opencv-demo-image.bb

```
DESCRIPTION = "robot-opencv-demo - Contains a basic X11 environment that
boots to a matchbox-terminal and allows you to run the OWI Robot Arm & OpenCV
MinnowBoard demo."

IMAGE_FEATURES += "splash package-management x11-base \
                   ssh-server-dropbear"

LICENSE = "MIT"

inherit core-image

IMAGE_INSTALL += "opencv-apps opencv-dev python-opencv python-modules \
                  python-pyusb python-wxpython mesa-demos"
```

Future Plans

- Use a python state machine library for application code
- Improve reliability, increase samples used in OpenCV training
- Add ability to detect fish by color
- Add control of fish color selection via the MinnowBoard's on-board GPIO pushbuttons

Resources

- MinnowBoard: http://minnowboard.org
- Yocto Project: http://yoctoproject.org
- OWI Robot Arm: http://www.owirobots.com/cart/html/owi-535-robotic-arm-edge-kit.html
- OpenCV: http://opencv.org
- Concise Explanation of Haar Classifiers: https://www.youtube.com/watch?v=0WBUIRADBd0
- Yocto Project Layer and Application Code: https://github.com/MinnowBoard

Thank you for your participation!