

360° Camera



using Odroid und Raspberry Pi

Agenda

The idea

Hardware

Realization

Conclusion

A panorama picture ...



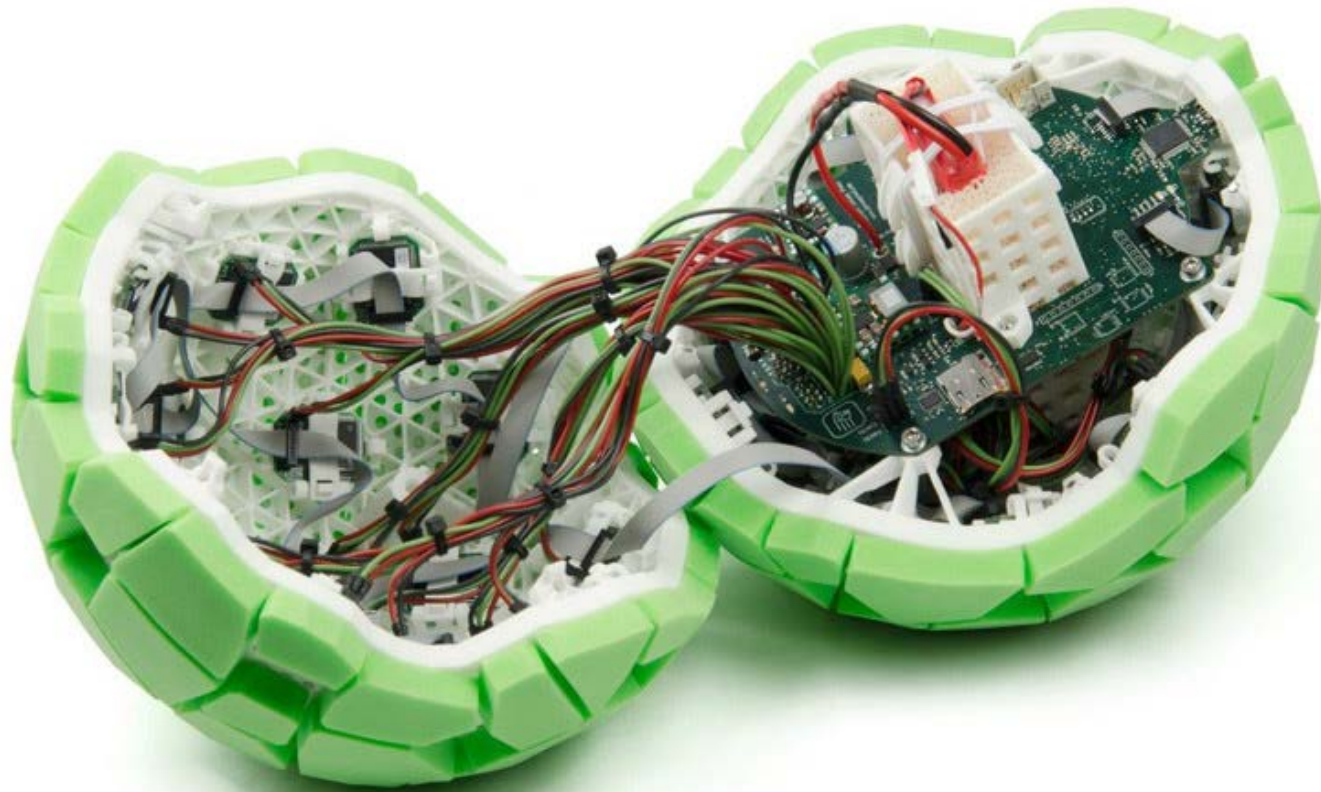
Schalzbackvorsäß, by Friedrich Böhringer

... in all directions



<http://occipital.com/static/newproduct/img/ex-stereo1.jpg>

The Panono camera



The Panono camera



<http://www.panono.com/ballcamera>

GoPro panorama camera



by Jonas Ginter

GoPro panorama camera

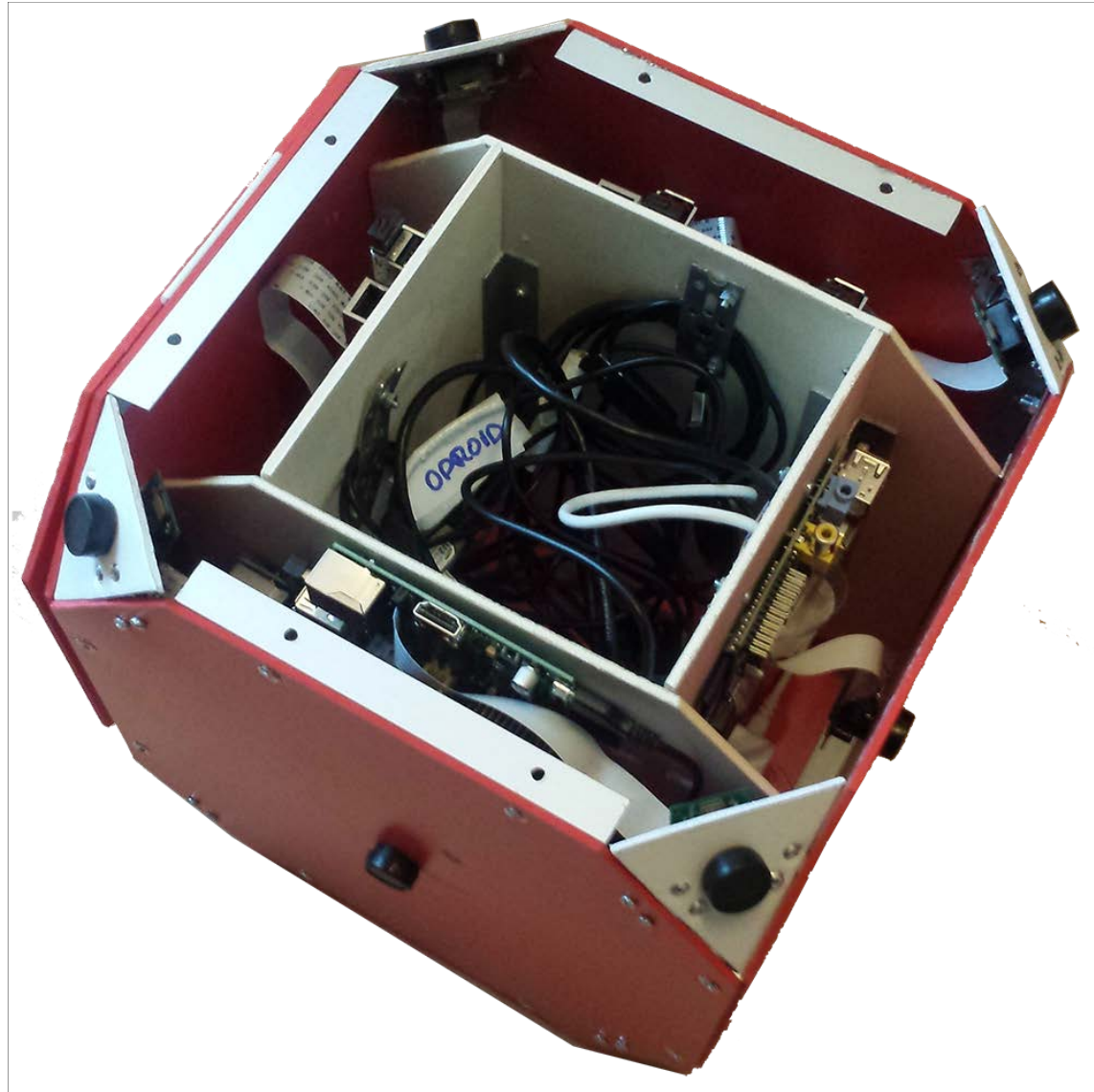


<http://www.jonasginter.de>

Virtual Surfers



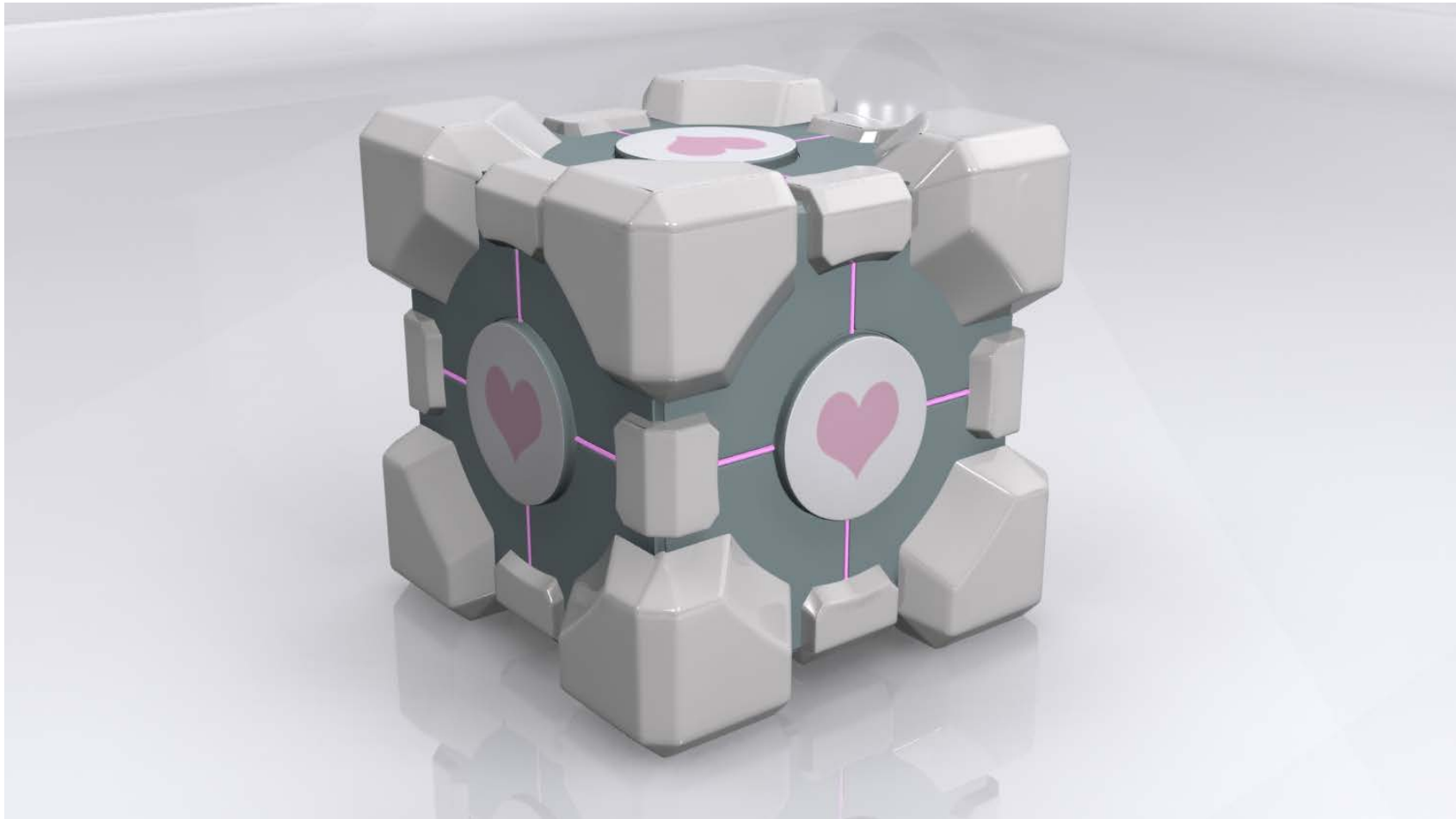
TNG Pano Cube



TNG Pano Cube

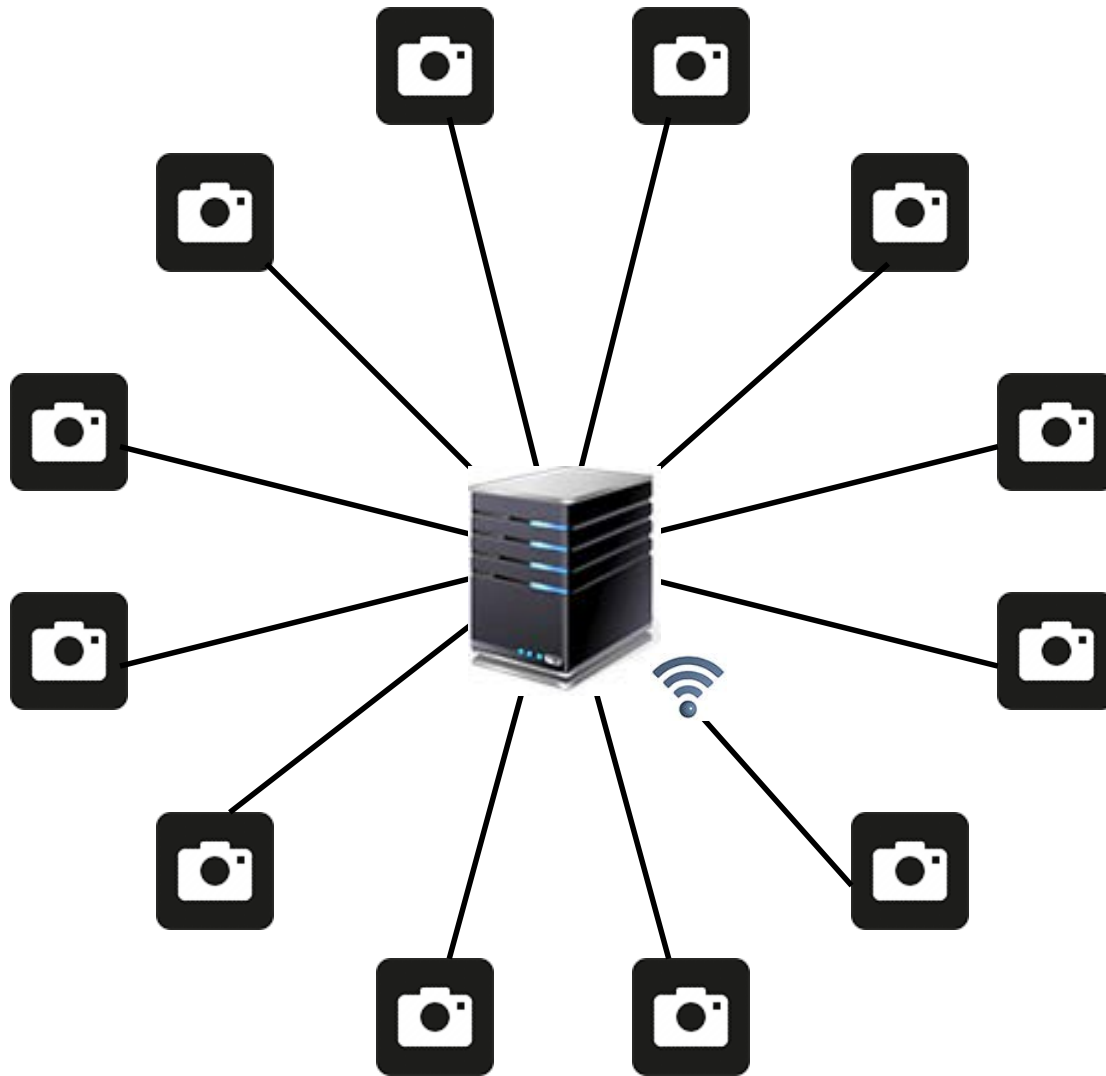


Why the heck are you doing it??



http://m.cdn.blog.hu/na/nanoretro/image/portal_weighted_companion_cube.jpg

Concept



Agenda

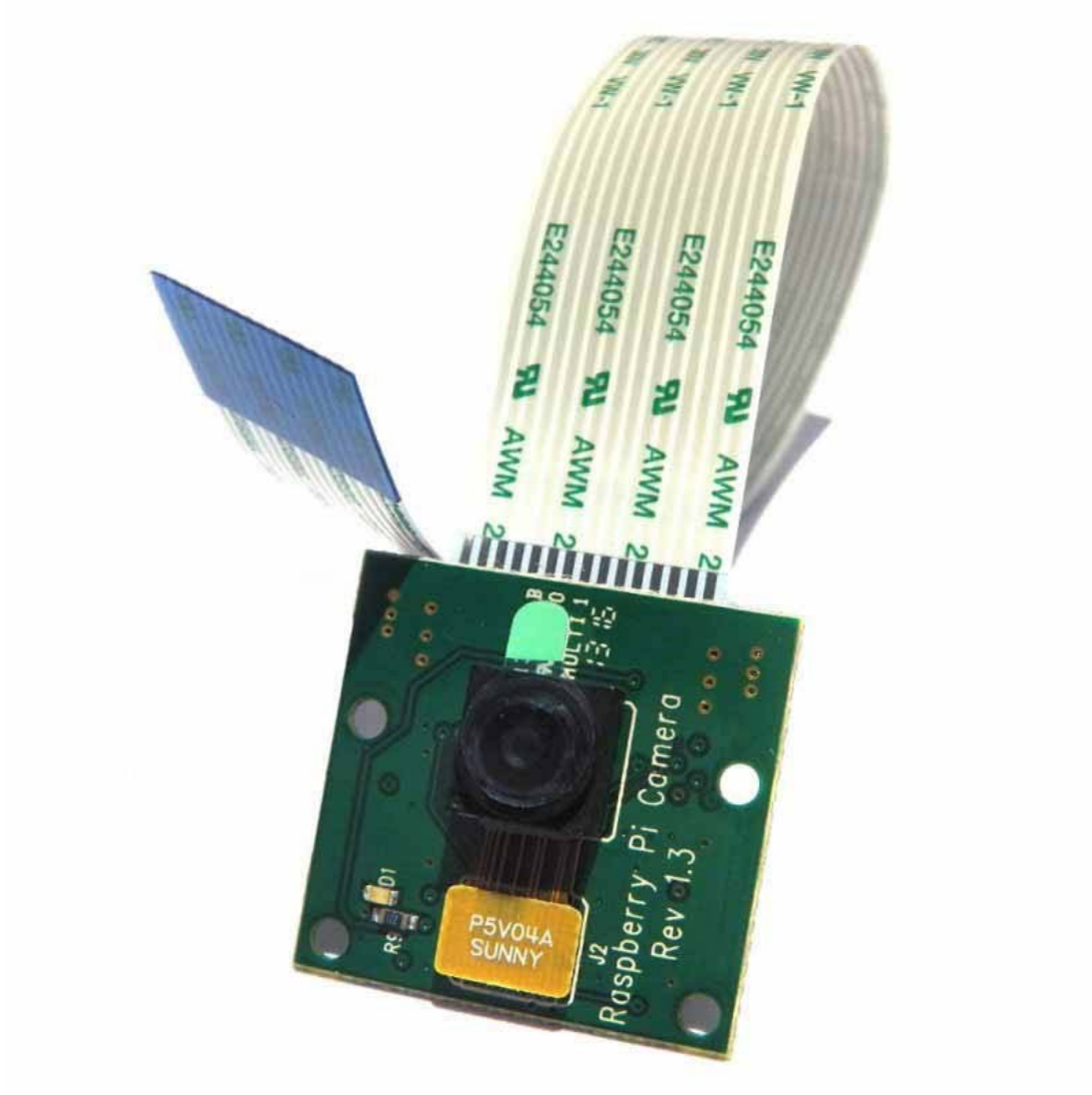
The idea

Hardware

Realization

Conclusion

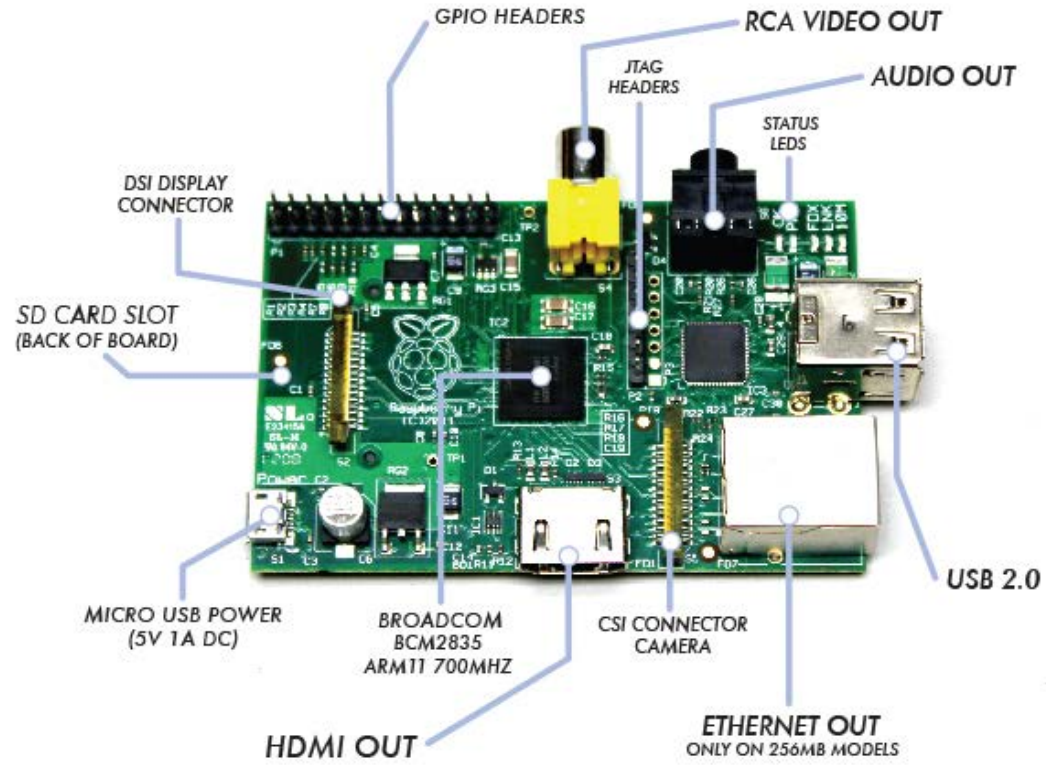
Raspberry Pi camera module



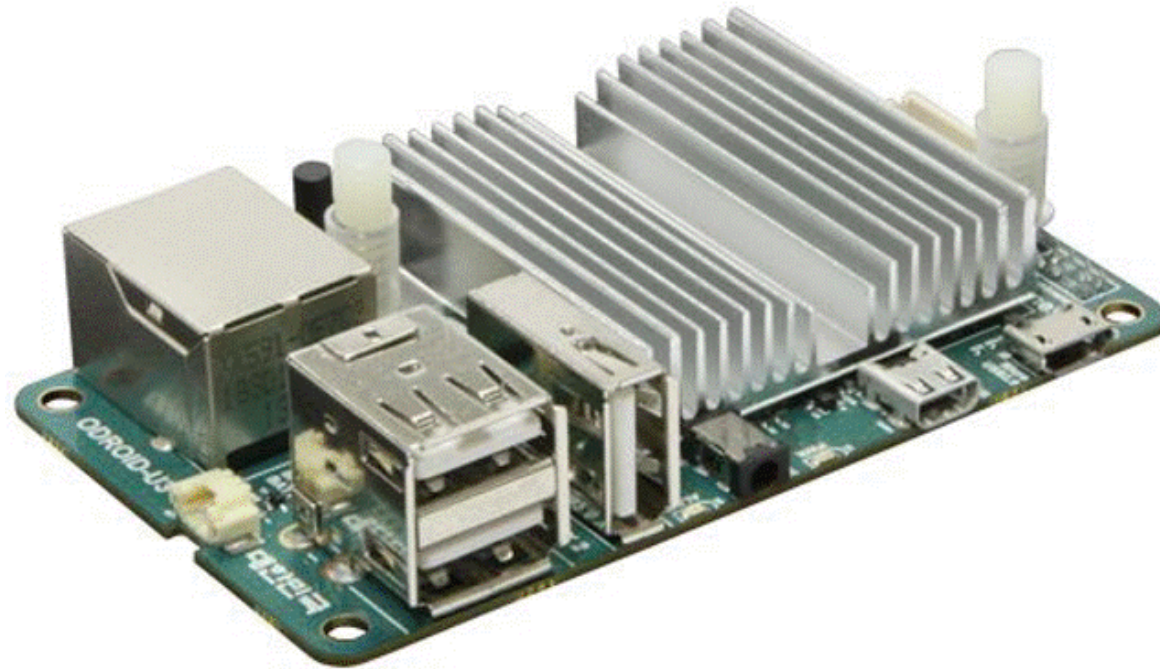
Raspberry Pi



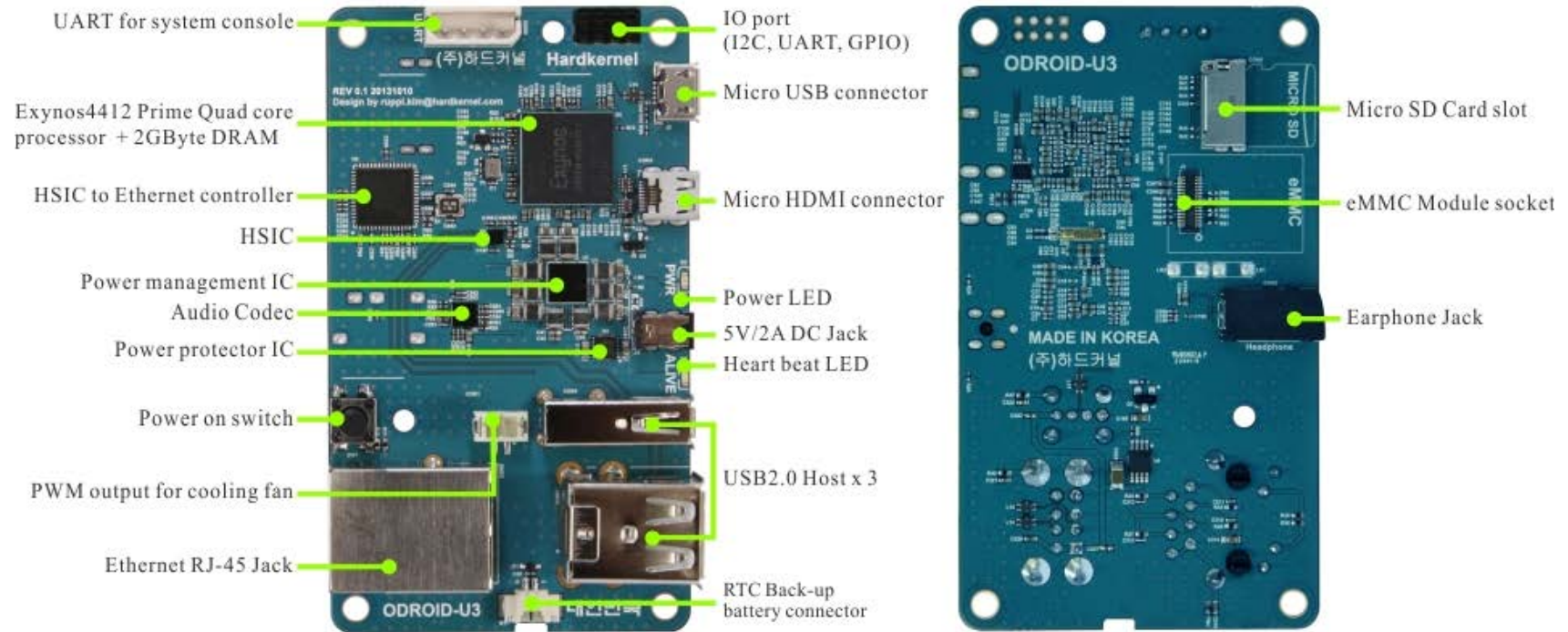
Raspberry Pi



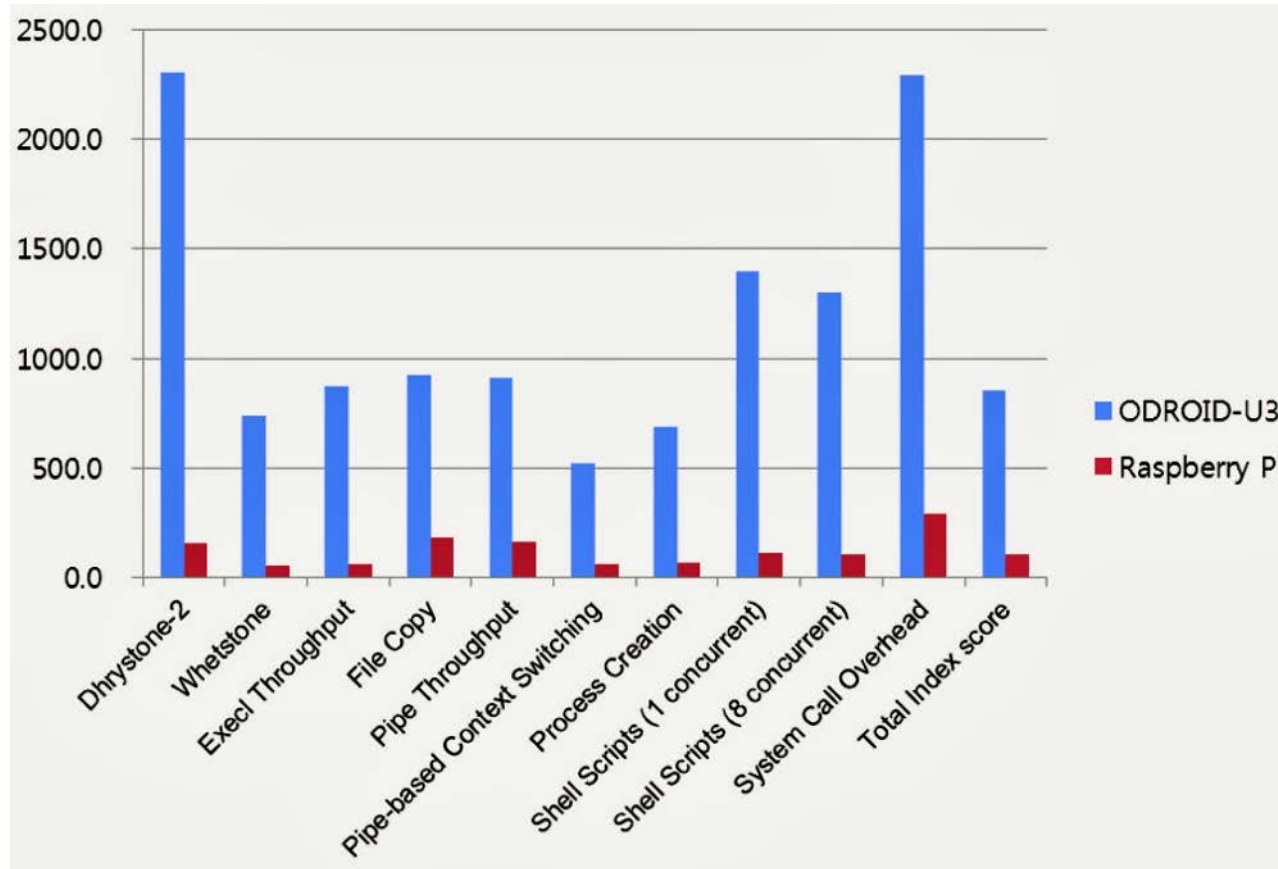
Odroid U3



Odroid U3



Performance Comparison



Source: Hardkernel, <http://linuxgizmos.com/open-sbc-runs-linux-on-quad-core-exynos-4412/>

Agenda

The idea

Hardware

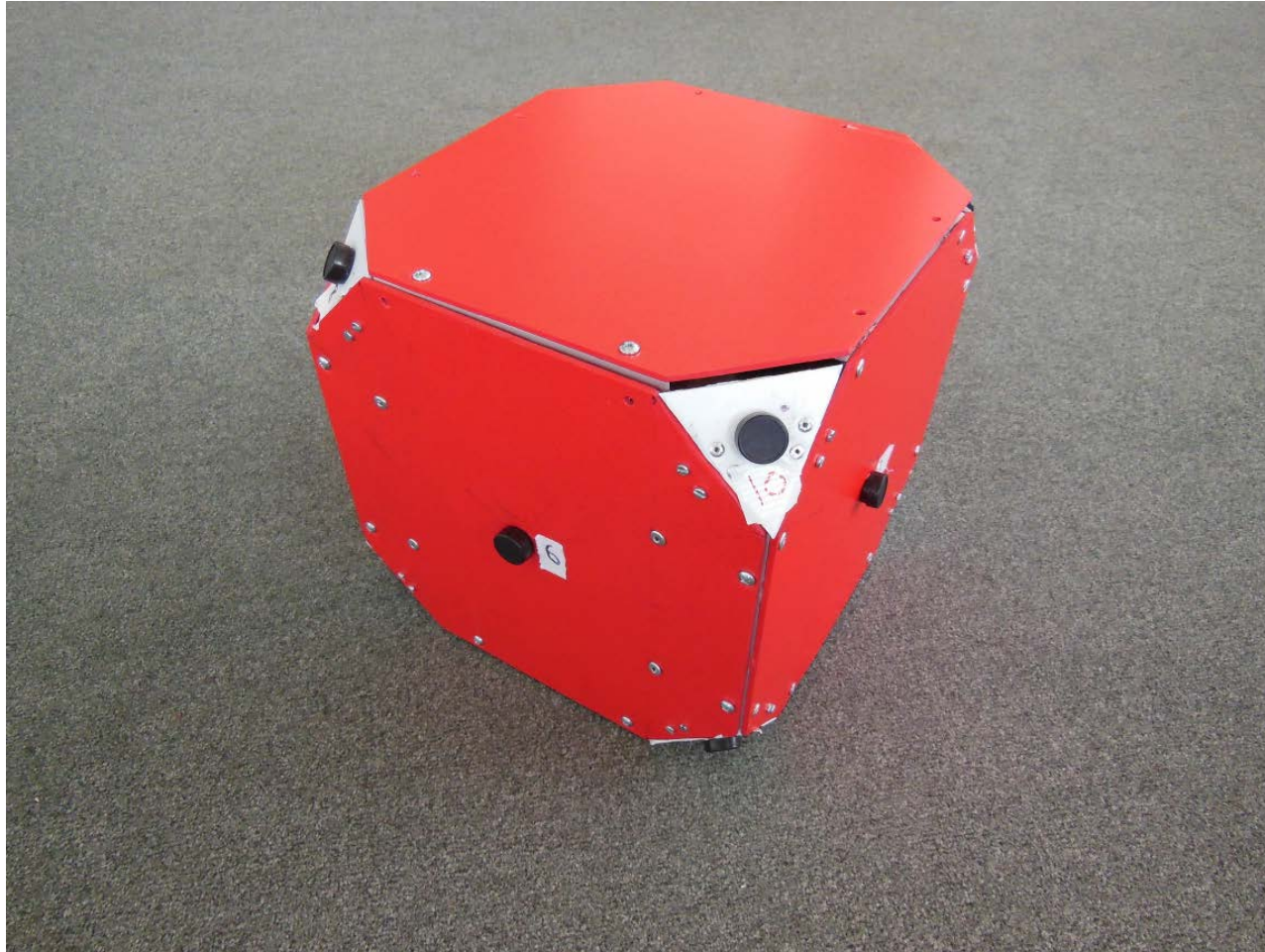
Realization

Conclusion

Concept - revisited



The Cube



Let's Focus on it



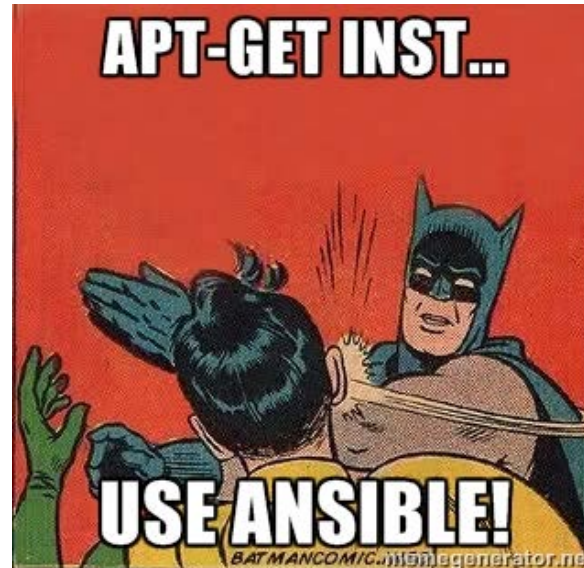
The Content



The Assembly



Provisioning



Inventory definition

```
[cam]
```

```
cam1
```

```
cam2
```

```
cam3
```

```
...
```

```
[odroid]
```

```
odroid
```

Playbooks



Playbook example

```
- hosts: cam
  sudo: True
  tasks:
  - lineinfile: dest=/etc/hosts line='192.168.1.99 odroid'
  - copy: src={{ inventory_dir }}/reconnect.sh dest=/usr/bin/reconnect.sh
        mode=0700
  - apt: pkg={{ item }} state=present
        with_items:
        - htop
        - nmap
  - cron: name='reconnect wifi' job='/usr/bin/reconnect.sh'
```

Ansible command line

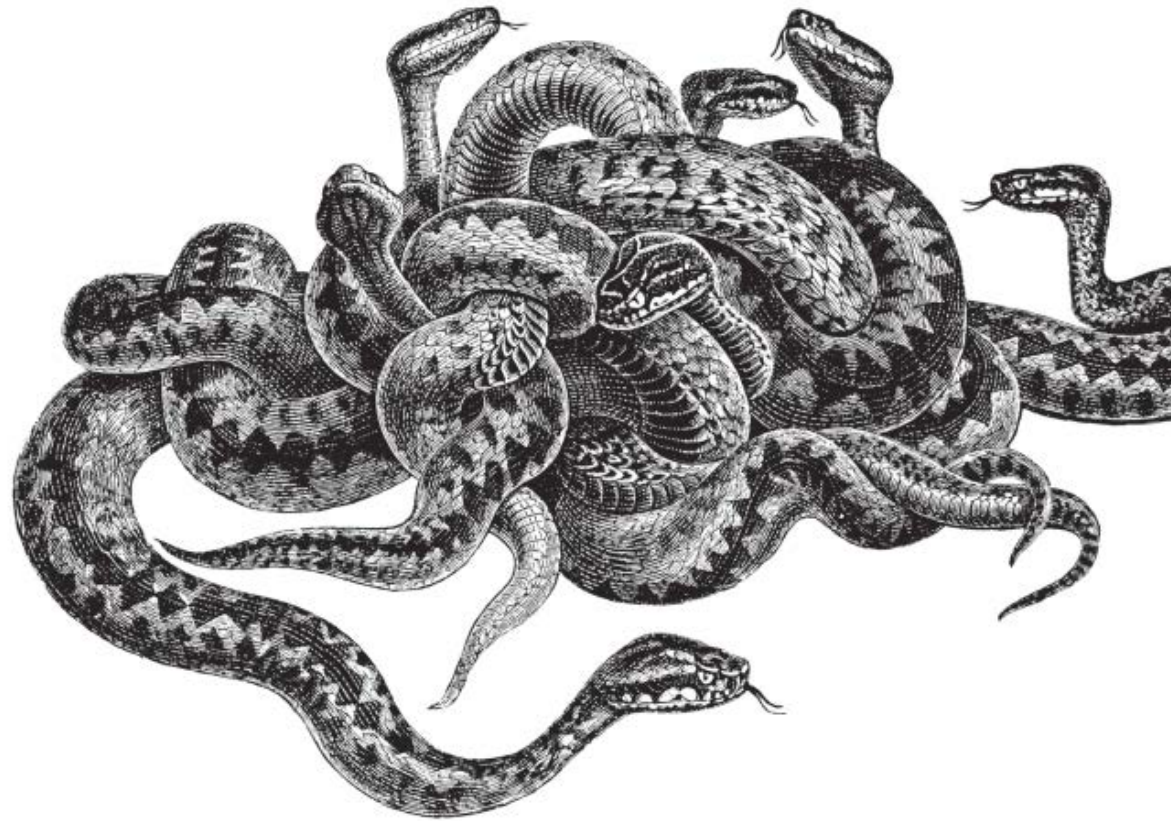
Execute remote commands:

```
# execute ping command as user bruce, sudoing to root  
$ ansible all -m ping -u bruce --sudo
```

Provision using a playbook:

```
# provision all cam hosts in inventory using playbook deploy.yml  
$ ansible-playbook -i inventory --limit cam deploy.yml
```

Data transfer



Twisted - Socket factory

```
class PanoFactory(protocol.Factory):
    def __init__(self):
        logging.info('Server started')

    def buildProtocol(self, addr):
        logging.info('New client at {}:{}'.format(addr.host, addr.port))
        return PanoProtocol(self)

if __name__ == "__main__":
    reactor.listenTCP(8100, PanoFactory())
    reactor.run()
```

Twisted - Socket client

```
class PanoProtocol(basic.LineReceiver):  
    def lineReceived(self, line):  
        logging.debug('Got "{}" from client'.format(line))  
  
    def sendCommand(self, command):  
        self.transport.write('{}\n'.format(command))  
  
    def connectionMade(self):  
        logging.info('Client connection made')  
  
    def connectionLost(self, reason):  
        logging.info('Client connection lost')
```

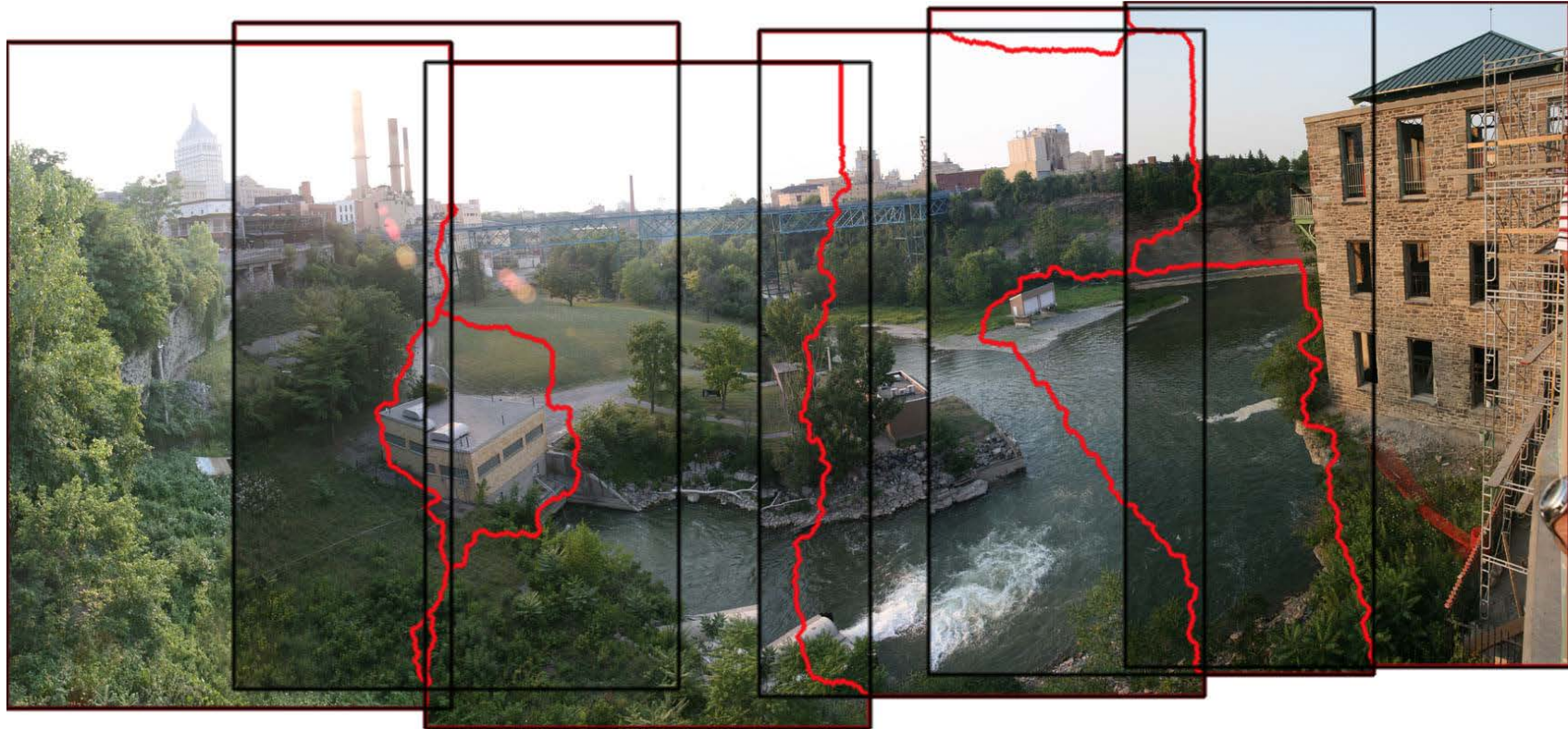
Trigger - Android



Stitching



Stage 1 - Registration



Stage 2 - Calibration



Stage 3 - Blending



Tools



Agenda

The idea

Hardware

Realization

Conclusion

Results



Results



Results



Results



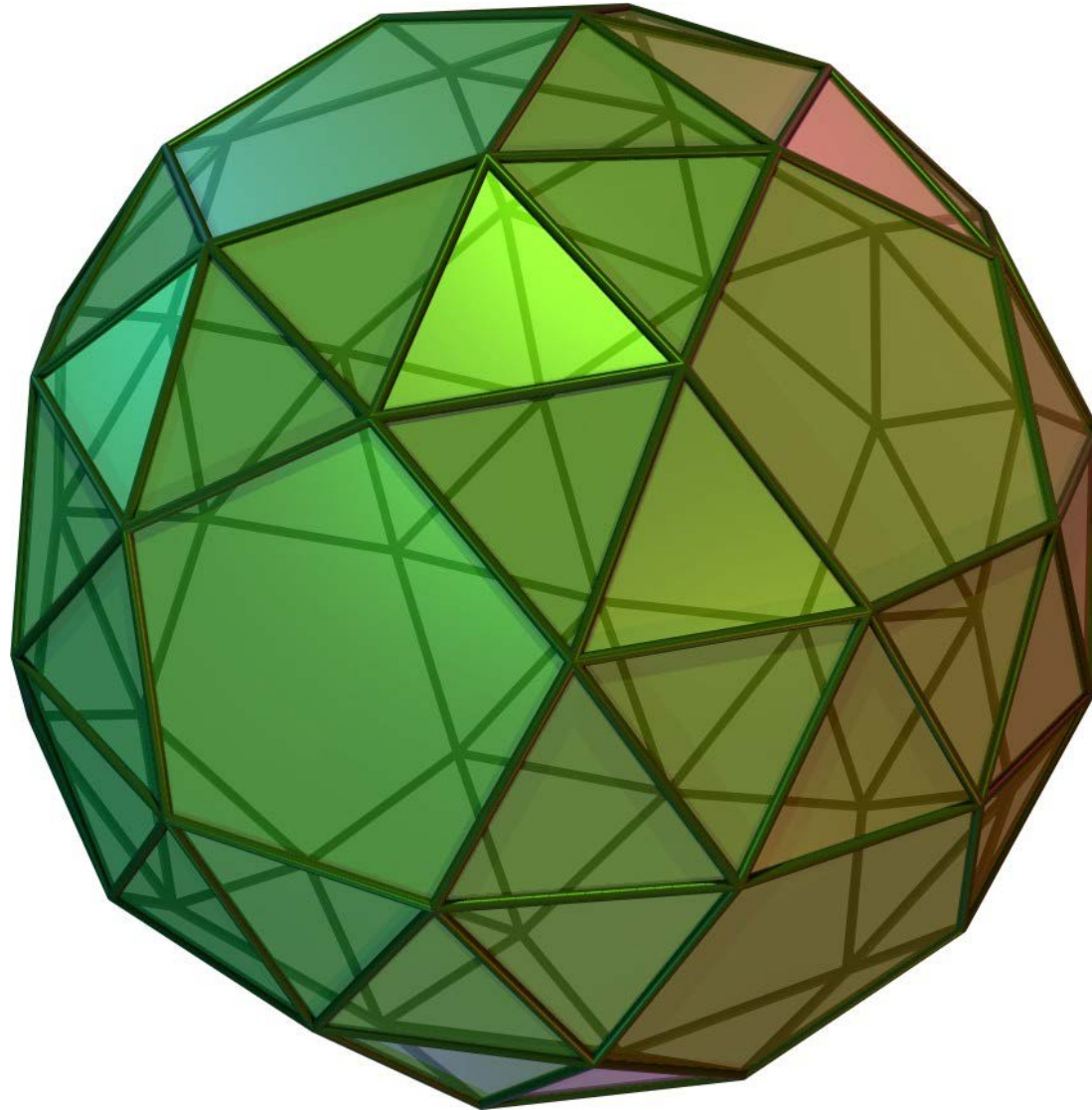
Improvements



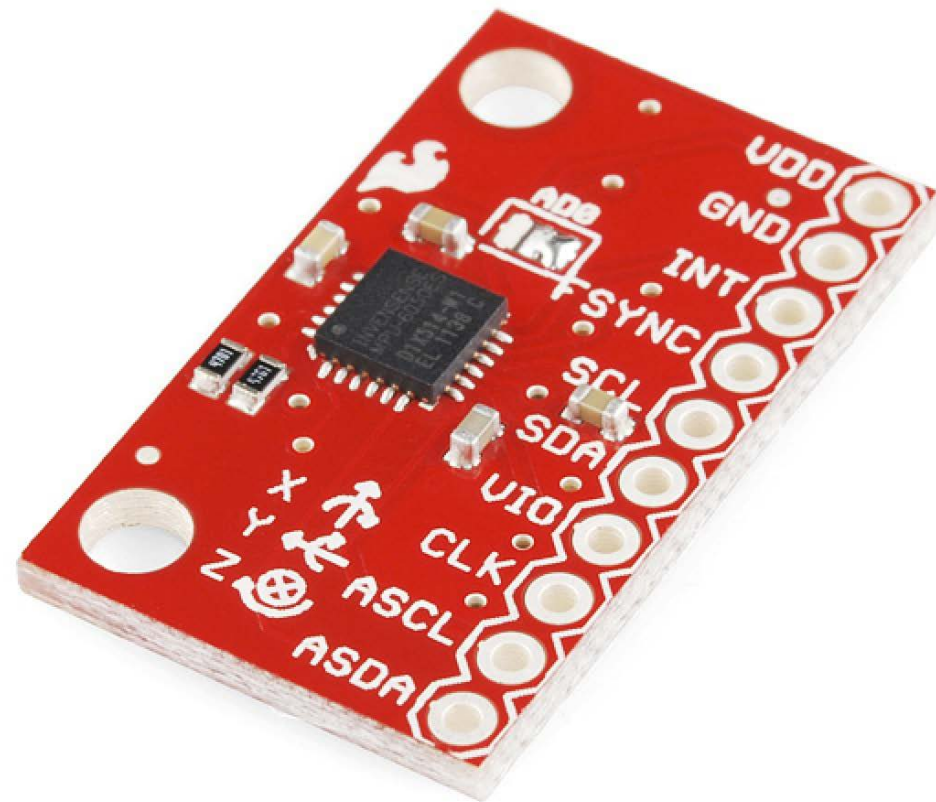
Improvements

E	1	20/200
F P	2	20/100
T O Z	3	20/70
L P E D	4	20/50
P E C F D	5	20/40
E D F C Z P	6	20/30
F E L O P Z D	7	20/25
D E F P O T E C	8	20/20
L E F O D P C T	9	
D P L T C E O	10	
F L O P T E D	11	

Improvements



Improvements





Andriy Samsonyuk



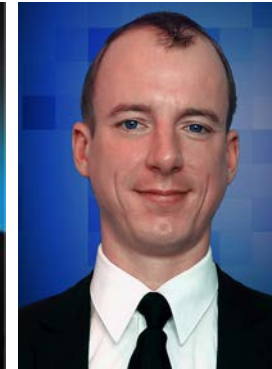
Daniele Fognini



Julian Exner



Manuel Meilinger



Martin Förtsch



Thomas Endres



Thomas Reifenberger



Woyten Tiesch