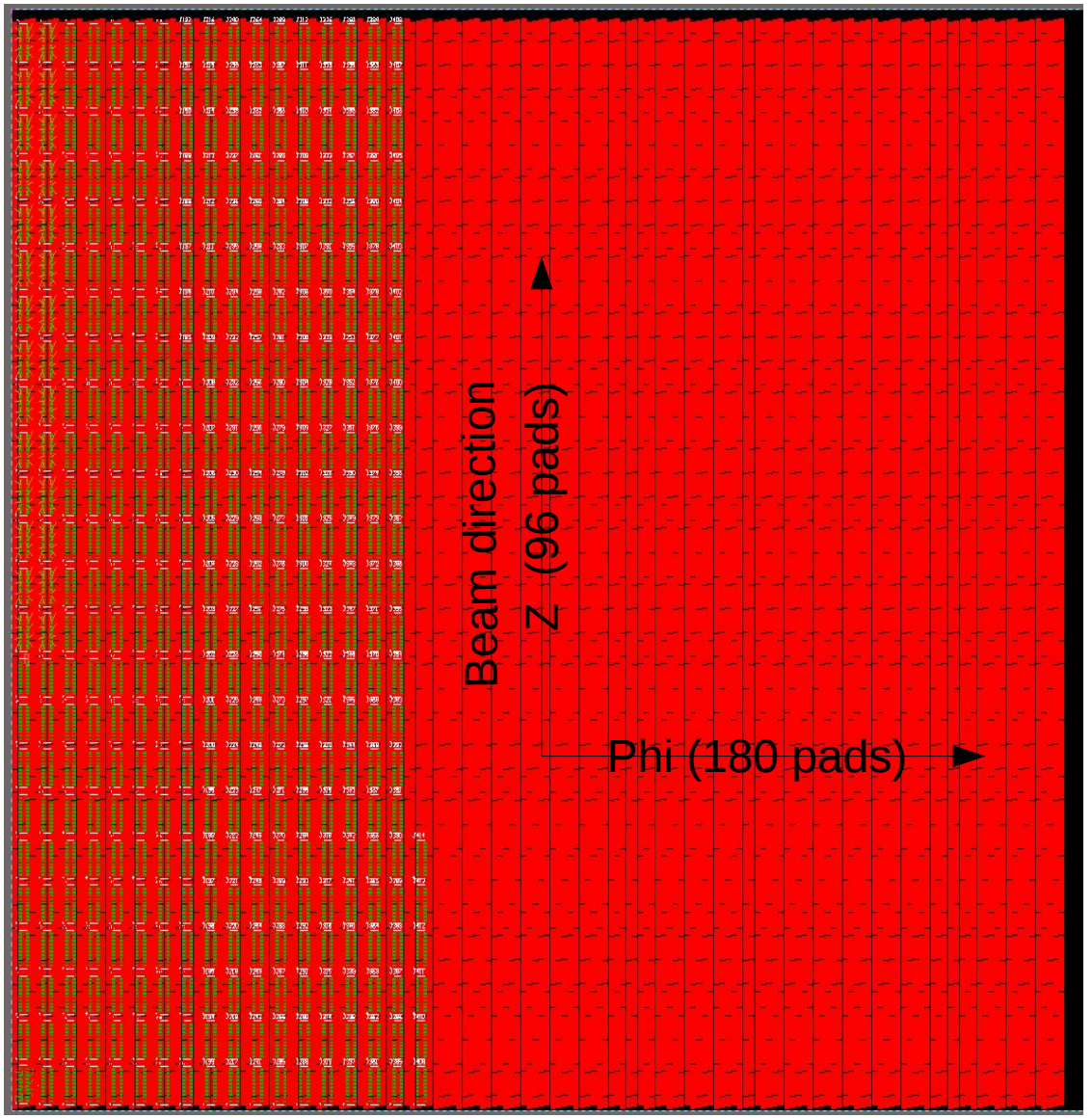
# RTPC12 Channel Map

Jixie Zhang

July 24nd, 2017

# Introduction

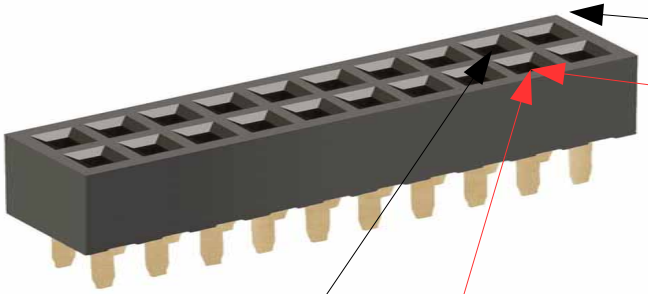
- RTPC pad PCB board will have 200 rows(phi) and 96 columns(z) of pad.
- Board size: 400mm(z) x 502mm, 10mil thick
- Pad size: 2.7mm x 4mm, 4 mil gap between pads
- Each connector will connect to 4x4 pads. There are 6 connectors in z direction, and 45 connector rows. In total there will be 1080 connectors.
- For these 4 rows of pads in each connector, there are z offsets for each row in order to provide z resolution for tracks with theta near 90 degrees. **(Has this shifting pattern been simulated and optimized?)**

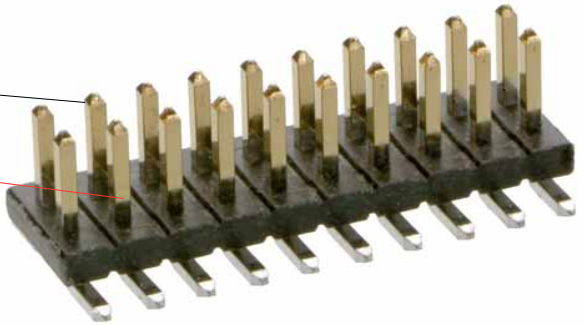


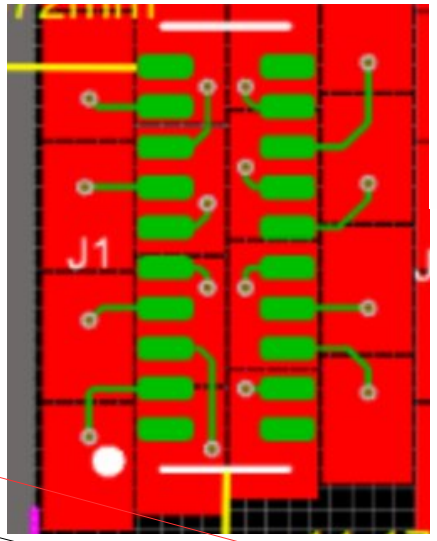
# Connector Layout

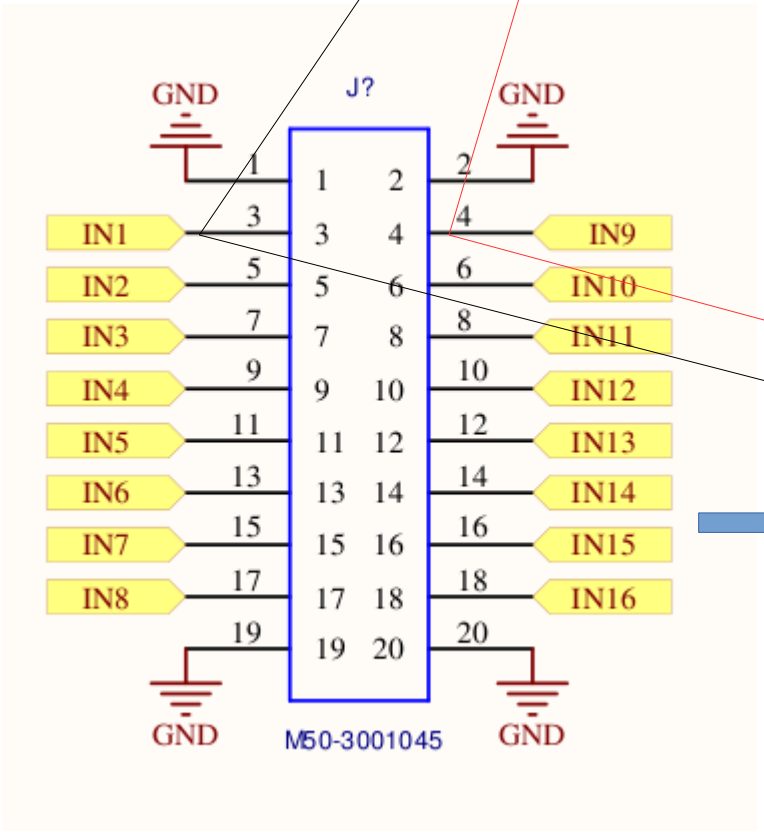
Female connector on the adapter board
M50-3001045

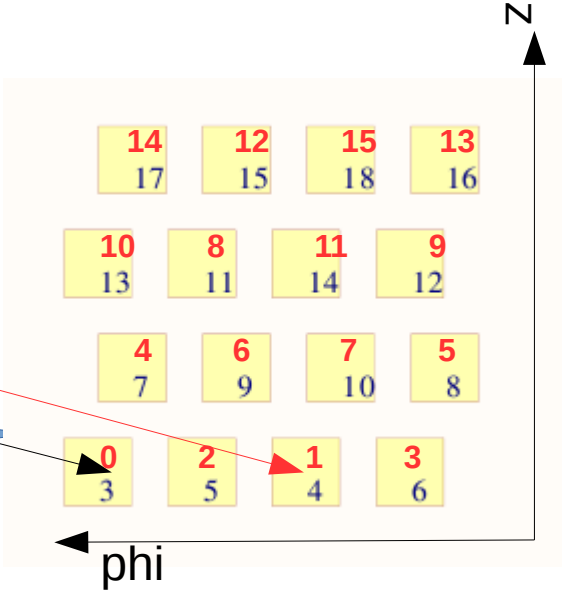
Male connector on the PCB board
M50-3601042R





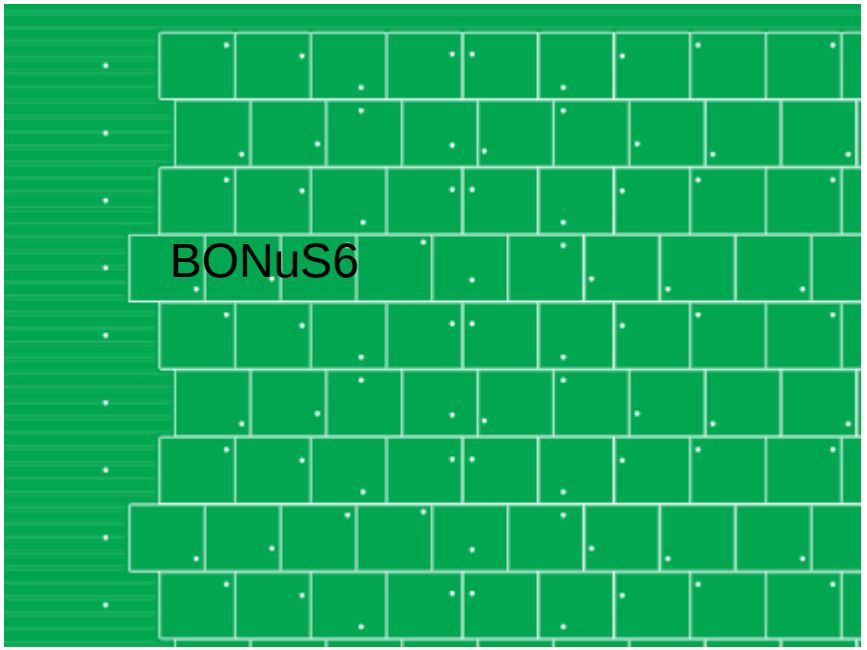


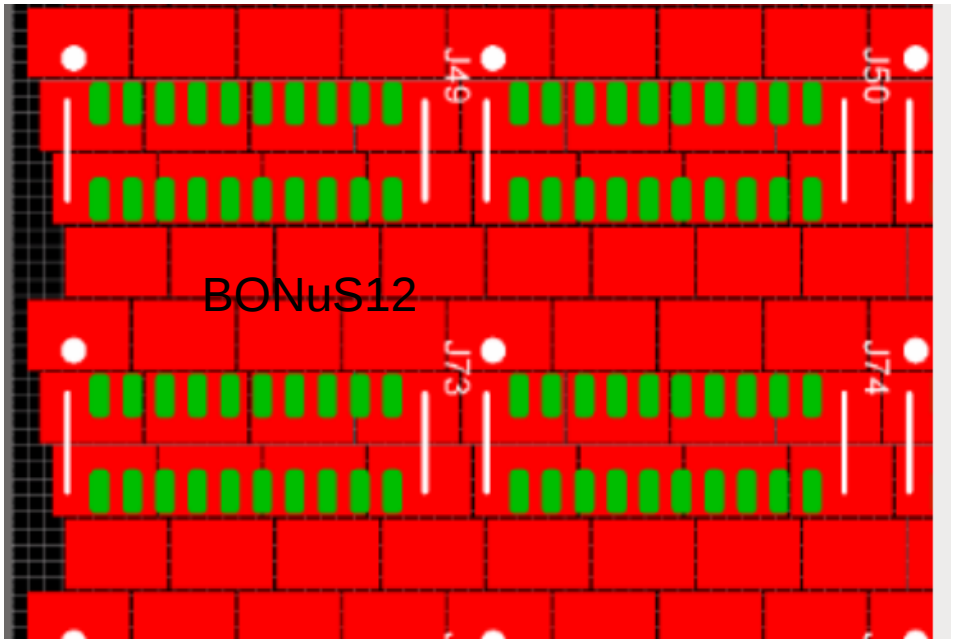




Black is pin index
Red is channel index

# Z offsets




Goal: provide z and theta resolution for tracks with theta near 90 degrees

Use a different pattern in BONuS12? Where does it come from? (EG6 design?)

Should we simulate this and optimize it?

These offsets is needed to build channel map.

# Channel Map Source Code Is Ready

**How to bulid it?**

```
//providing pad width(phi), length(z), inner_s and
//phi at the bottom edge of the connector connected to channel 0

ChannelMap(double pad_w=2.7381, double pad_l=4.1016, double pad_r_in=79.0,
           double phi_conn_edge=0.);

//Z offsets need to verify to final design
static const double kZOffset[4]= { 1.8, 1.2, 0.6, 0.0};
```

**How to use it?**

```
double GetZ(int chan);    //return the vertex Z position in mm
double GetPhi(int chan);  //return the Phi angle in rad
void     GetZPhi(int chan, double &z_mm, double &phi_rad);

int        GetPadRowNCol(int pChan,int &row,int &col);
int        GetPadRowNCol(double z_mm, double phi_rad, int &row, int &col);

int        GetPadID(double z_mm,double phi_rad);
int        GetPadIDByRowNCol(int row, int col);
```

# Channel Map Snapshot

Creating Channel map: 180 Rows x 96 Columns = 17280 pads in total
Pad Width(2.7381mm) Length(4.1016mm) Inner_S(79.000mm) fDeltaPhi(1.986deg)
fPhiStart(0.000deg) fZStart[4]={-195.077, -195.677, -196.277, -196.877} (mm)

| ID | z_mm | phi_deg | row | col |
|---|---|---|---|---|
| 0 | -194.83 | 6.95 | 3 | 0 |
| 1 | -193.63 | 2.98 | 1 | 0 |
| 2 | -194.23 | 4.96 | 2 | 0 |
| 3 | -193.03 | 0.99 | 0 | 0 |
| 4 | -190.72 | 6.95 | 3 | 1 |
| 5 | -188.92 | 0.99 | 0 | 1 |
| 6 | -190.12 | 4.96 | 2 | 1 |
| 7 | -189.52 | 2.98 | 1 | 1 |
| 8 | -186.02 | 4.96 | 2 | 2 |
| 9 | -184.82 | 0.99 | 0 | 2 |
| 10 | -186.62 | 6.95 | 3 | 2 |
| 11 | -185.42 | 2.98 | 1 | 2 |
| 12 | -181.92 | 4.96 | 2 | 3 |
| 13 | -180.72 | 0.99 | 0 | 3 |
| 14 | -182.52 | 6.95 | 3 | 3 |
| 15 | -181.32 | 2.98 | 1 | 3 |

| ID | z_mm | phi_deg | row | col |
|---|---|---|---|---|
| 16 | -178.42 | 6.95 | 3 | 4 |
| 17 | -177.22 | 2.98 | 1 | 4 |
| 18 | -177.82 | 4.96 | 2 | 4 |
| 19 | -176.62 | 0.99 | 0 | 4 |
| 20 | -174.32 | 6.95 | 3 | 5 |
| 21 | -172.52 | 0.99 | 0 | 5 |
| 22 | -173.72 | 4.96 | 2 | 5 |
| 23 | -173.12 | 2.98 | 1 | 5 |
| 24 | -169.62 | 4.96 | 2 | 6 |
| 25 | -168.42 | 0.99 | 0 | 6 |
| 26 | -170.22 | 6.95 | 3 | 6 |
| 27 | -169.02 | 2.98 | 1 | 6 |
| 28 | -165.51 | 4.96 | 2 | 7 |
| 29 | -164.31 | 0.99 | 0 | 7 |
| 30 | -166.11 | 6.95 | 3 | 7 |
| 31 | -164.91 | 2.98 | 1 | 7 |

https://github.com/jixie/AnaRTPC/blob/master/src/ChannelMap.hh
https://github.com/jixie/AnaRTPC/blob/master/src/ChannelMap.cc

# Channel Map Source Code Is Ready

```
//providing pad width(phi), length(z), inner_s, phi at center of the
//bottom edge of the connector connected to channel 0
ChannelMap(double pad_w=2.7381, double pad_l=4.1016, double pad_r_in=79.0,
           double phi_conn_edge=0.);

//Z offsets need to verify to final design
static const double kZOffset[4]= { 1.8, 1.2, 0.6, 0.0};
```

Output:

```
double GetZ(int chan);    //return the vertex Z position in mm
double GetPhi(int chan);  //return the Phi angle in rad
void     GetZPhi(int chan, double &z_mm, double &phi_rad);

int        GetPadRowNCol(int pChan,int &row,int &col);
int        GetPadRowNCol(double z_mm, double phi_rad, int &row, int &col);

int        GetPadID(double z_mm,double phi_rad);
int        GetPadIDByRowNCol(int row, int col);
```

# Channel Map Source Code Is Ready

```
//providing pad width(phi), length(z), inner_s, phi at center of the
//bottom edge of the connector connected to channel 0
ChannelMap(double pad_w=2.7381, double pad_l=4.1016, double pad_r_in=79.0,
           double phi_conn_edge=0.);

//Z offsets need to verify to final design
static const double kZOffset[4]= { 1.8, 1.2, 0.6, 0.0};
```

Output:

```
double GetZ(int chan);    //return the vertex Z position in mm
double GetPhi(int chan);  //return the Phi angle in rad
void     GetZPhi(int chan, double &z_mm, double &phi_rad);

int        GetPadRowNCol(int pChan,int &row,int &col);
int        GetPadRowNCol(double z_mm, double phi_rad, int &row, int &col);

int        GetPadID(double z_mm,double phi_rad);
int        GetPadIDByRowNCol(int row, int col);
```

# Summary

1. The design of the channel map should only favor the DAQ such that it run efficiently and reduce dead time

2. Need to verify the final design for the following: gap between pads, inner radius of the pad PCB board, dead region, Zoffsets.

3. Z shifted pattern might need to be optimized. If yes, this has to be done ASAP.

4. source code to build channel map is ready to use.