Auto Bed Leveling using Inductive Sensor, Buck converter, and Optocoupler

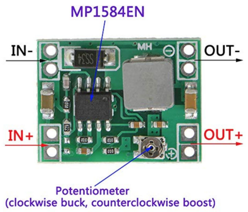
(For Monoprice Maker Select Plus and similar 3D Printers)

I have implemented this on my Monoprice IIIP but the concept itself would also work on other kinds of 3D printers as well. Even if you do not have the same type of sensor or buck converter or optocoupler as I did. But of course, always proceed WITH CAUTION especially if you do not have the exact components because deviations from what I had done also require slight changes on wiring or firmware edits (which I cannot cover on this document anymore).

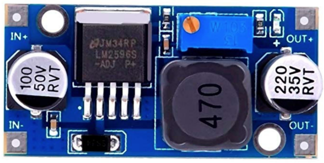
Materials:

- Inductive Sensor LJ12A3-4-Z/BY (PNP; Normally Open)

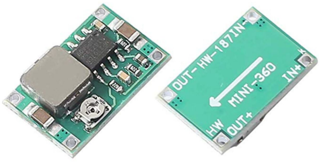
- Buck Converter (MP1584 or LM2596 or Mini360)



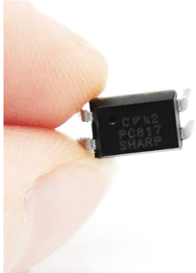
Or



Or



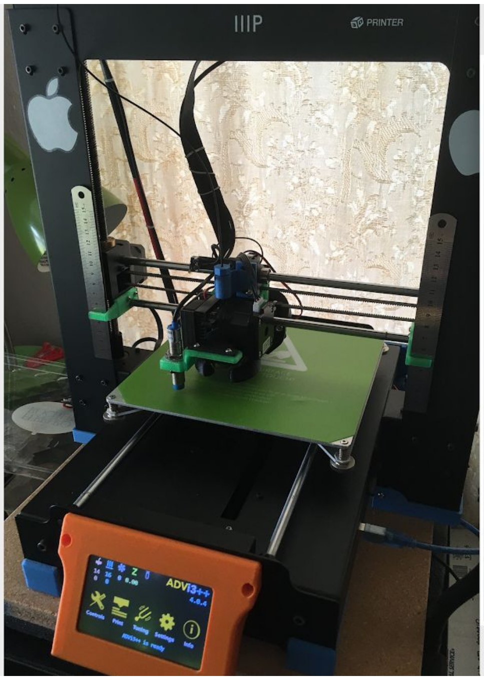
- Optocoupler (PC817) or similar variants

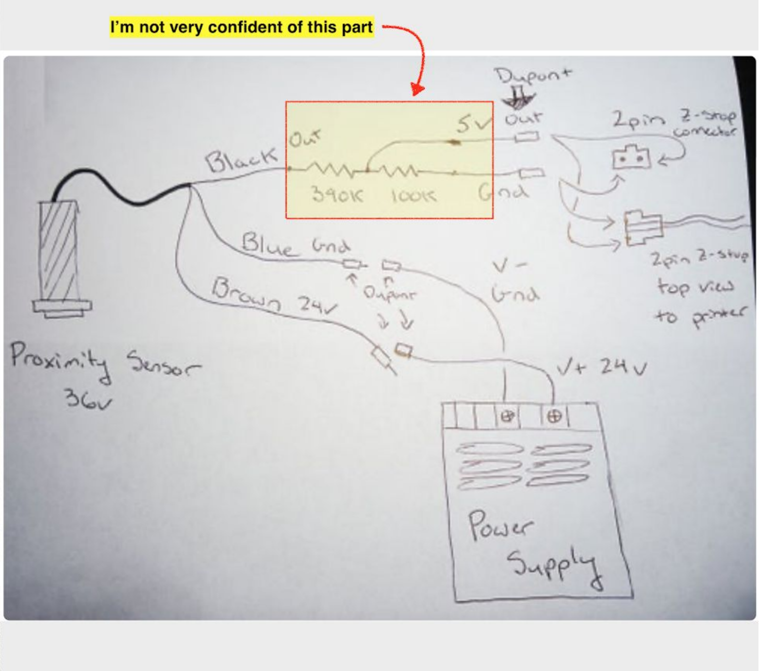


- JST SM Connectors
  - 3 pin
  - 2 pin

<u>Background:</u>

I currently have my inductive sensor installed and is already working. I used the instructions in this link (https://www.instructables.com/id/Wanhao-I3-Plus-Bed-Auto-Leveling-W-Inductive-Senso/). So basically, my setup is using this diagram…..





I'm not very confident of this part

However, having a voltage divider with ¼ watt resistors directly connect to the endstop connector (which directly connects to the board) makes me uneasy. If something wrong happens to the voltage divider, high voltage may go directly to the mainboard and break it.

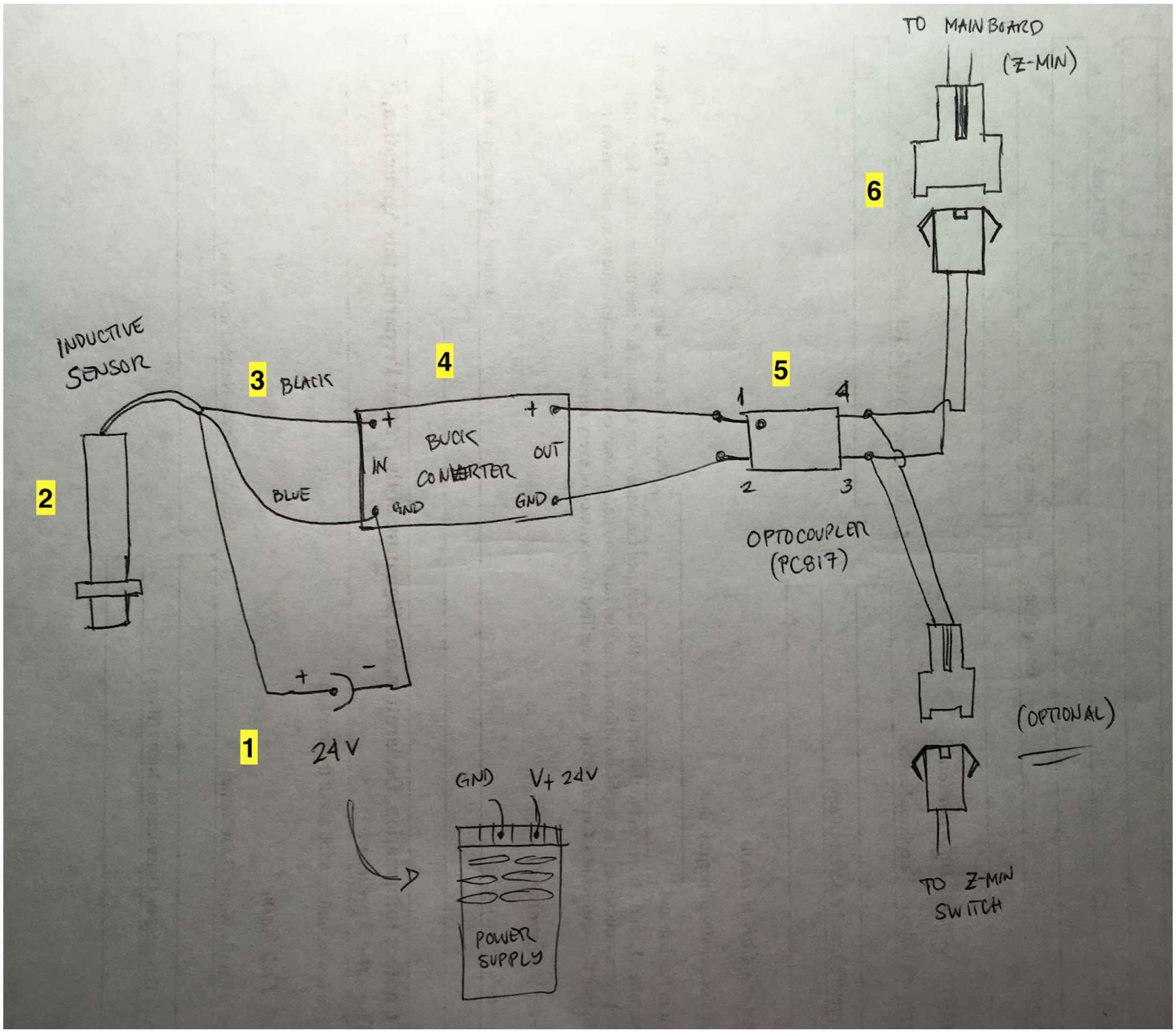
Objectives:

- Protect the mainboard from any possible problems from the sensor
- Very minimal need to open and solder connectors to the mainboard
- Use whatever items I already have lying around

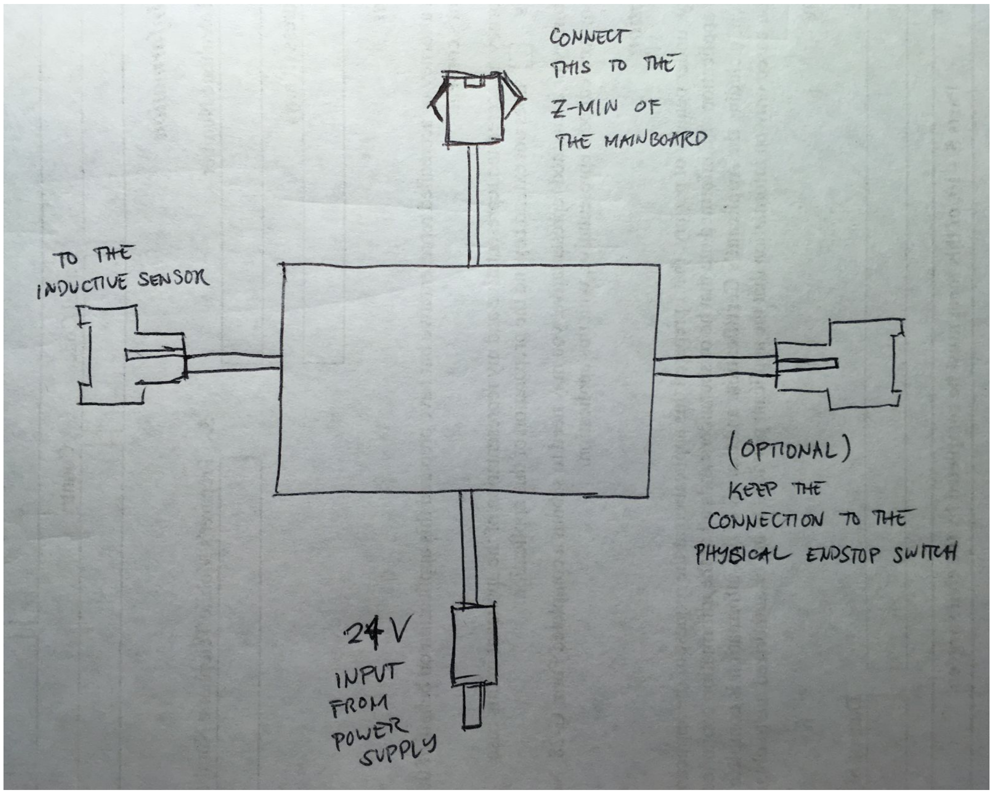
Expected Solution:

1. Printer's power supply provides power to the sensor. If you do not want to open up your printer to get power from the power supply of your board, you can actually use other sources. I'm assuming that would work too -- but you might forget to turn it on when you turn on your printer :)
2. Sensor detects metal bed
3. Sensor sends voltage to buck converter (voltage would be equivalent to voltage provided by power supply)
4. Buck converter steps down this voltage to a CONSTANT 1.4v DC -- because this is what the optocoupler will accept without complaining :)
5. Optocoupler, sensing the 1.4v from the buck converter from pins 1 and 2, would allow current to pass through pins 3 and 4
6. Z-min endstop gets triggered as a result of the current passing through the optocoupler

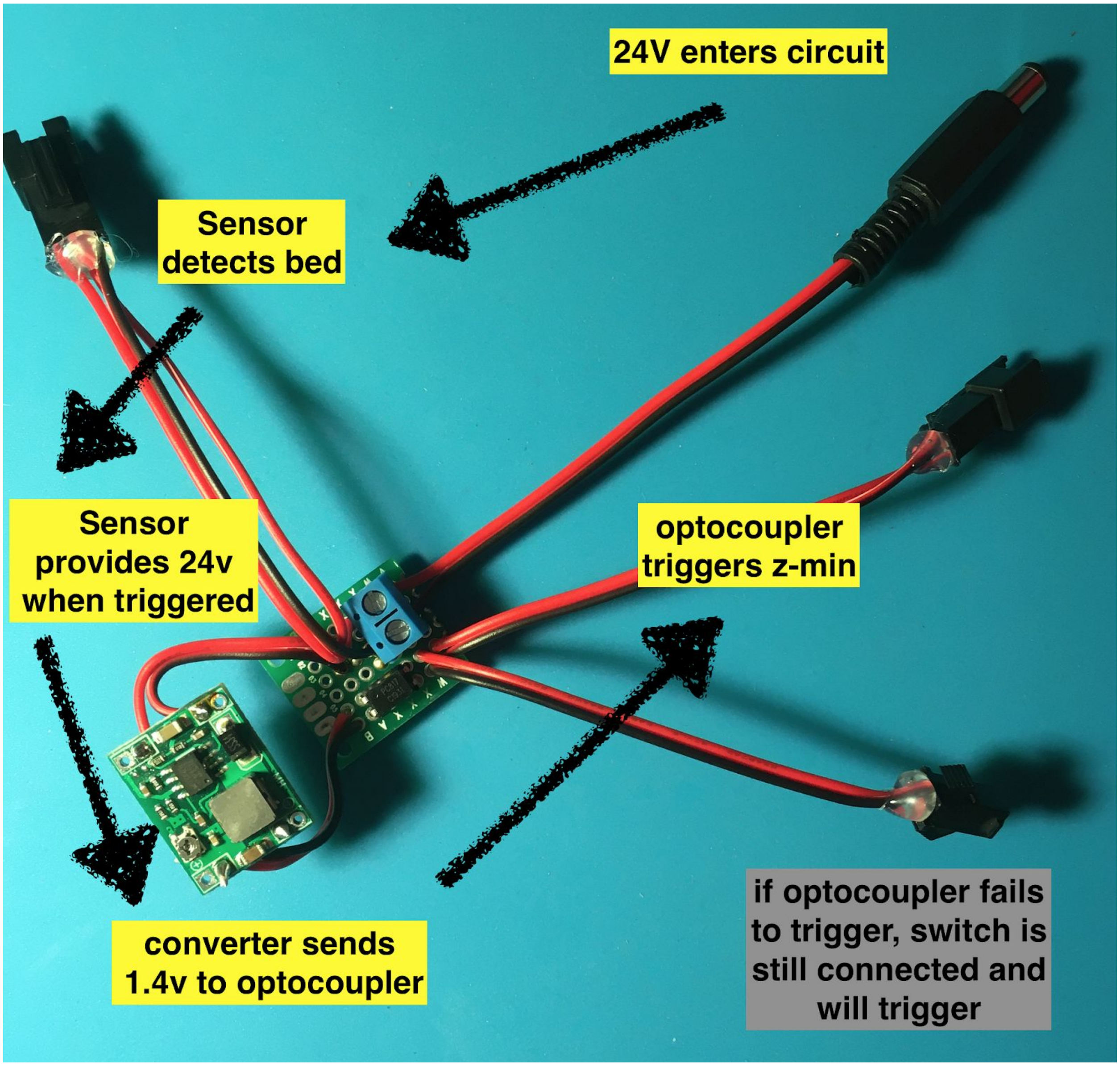
Refer to the diagram below:



Therefore, let’s build something like this….. (input power can be anything between 6 to 36volts).

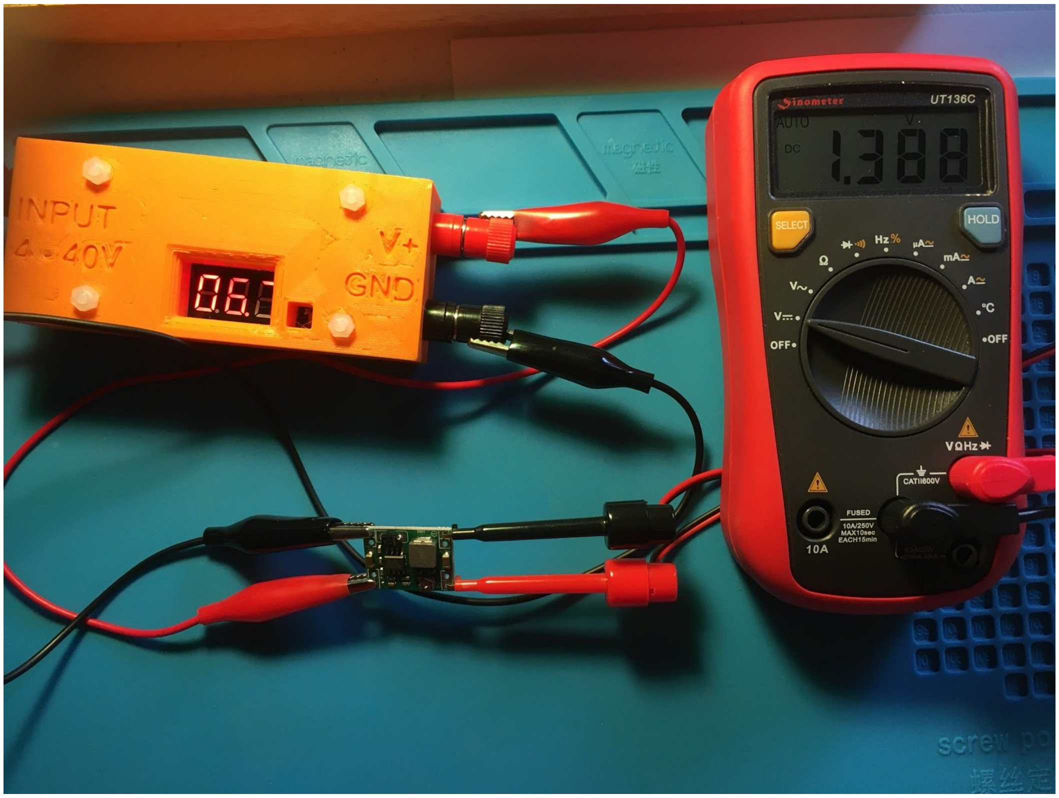


Resulting module:


24V enters circuit
Sensor detects bed
Sensor provides 24v when triggered
optocoupler triggers z-min
converter sends 1.4v to optocoupler
if optocoupler fails to trigger, switch is still connected and will trigger

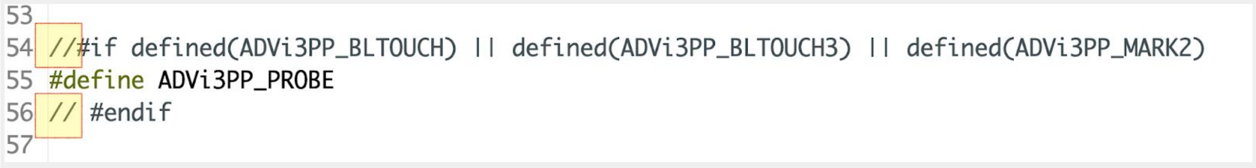
VERY IMPORTANT:

1. **Before you connect your buck converter to optocoupler, make sure the tiny potentiometer is already set to an output of 1.4 volts.**
    a. Connect external power to input terminals (anything more than 3 volts)
    b. Connect multimeter to output terminals
    c. Rotate potentiometer until multimeter reads 1.4v output



2. **Before you connect your circuit to your printer, modify firmware first (Marlin version of ADVi3++)**
    a. Download Marlin firmware from https://github.com/andrivet/ADVi3pp
    b. **It is highly recommended that you support the developer of ADVI3++ by becoming a patreon.**
    c. Using Arduino IDE (or however you choose to update your firmware), edit the following 3 files and then upload entire firmware to your board.

advi3pp_defines.h — disabled the #if and #endif



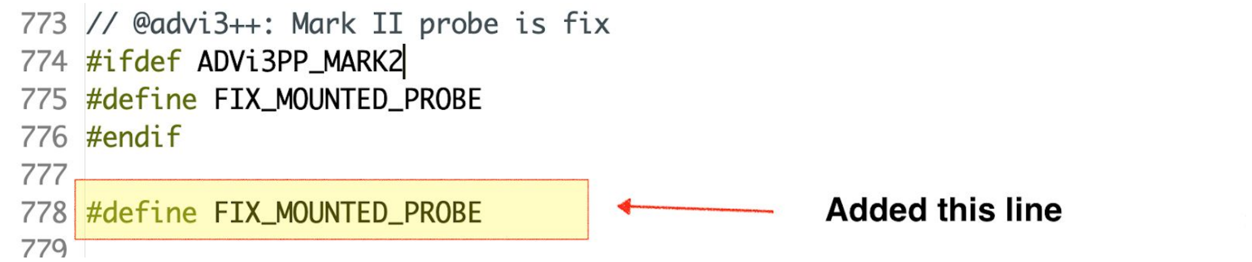
```

//#if defined(ADVi3PP_BLTOUCH) || defined(ADVi3PP_BLTOUCH3) || defined(ADVi3PP_MARK2)
#define ADVi3PP_PROBE
// #endif

```

Configuration.h

— defined fix mounted probe outside of the #ifdef / #endif block of code because it only gets defined for mark 2 builds

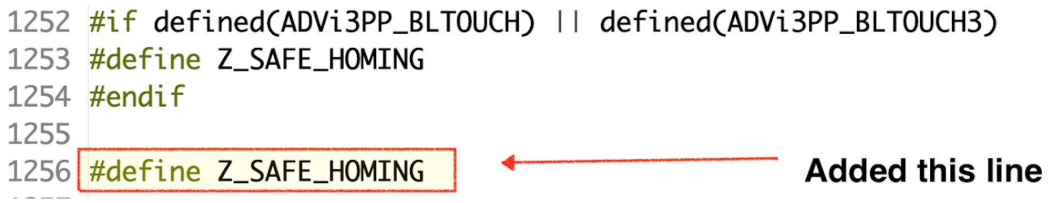


```
// @advi3++: Mark II probe is fix
#ifdef ADVi3PP_MARK2
#define FIX_MOUNTED_PROBE
#endif

#define FIX_MOUNTED_PROBE

```

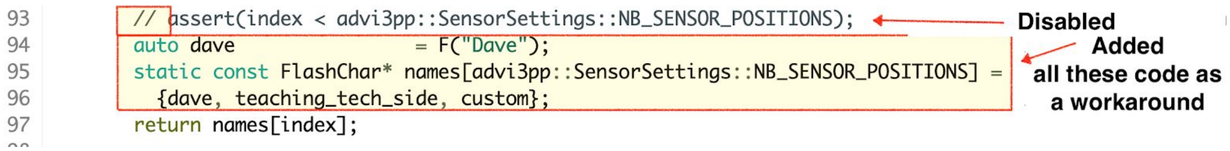
— similar to above, defined safe homing because it only gets defined for bltouch builds



```
#if defined(ADVi3PP_BLTOUCH) || defined(ADVi3PP_BLTOUCH3)
#define Z_SAFE_HOMING
#endif

#define Z_SAFE_HOMING
```

advi3pp_handlers.cpp – remarked the assert line and created my own holder



```
    // assert(index < advi3pp::SensorSettings::NB_SENSOR_POSITIONS);
    auto dave                 = F("Dave");
    static const FlashChar* names[advi3pp::SensorSettings::NB_SENSOR_POSITIONS] =
      {dave, teaching_tech_side, custom};
    return names[index];
```