EV-ADE9153A SHIELDZ Code corrections by Robert46 – 21-10-2018

1) Arduino Code ADE9153AAPI_Test.ino

During calibration, the code calls *routine runLength(n)* where n is the required delay in seconds as shown below:

```
void runLength(int seconds)
{
 unsigned long startTime = millis();

 while (millis() - startTime < (seconds*1000)){
  digitalWrite(LED, HIGH);
  delay(blinkInterval);
  digitalWrite(LED, LOW);
  delay(blinkInterval);
 }
}
```

Variable *seconds* is declared as *int* and subsequently used in a comparison between its value * 1000 and the difference between 2 *unsigned long* variables.
When called with a value > 32, the product (seconds*1000) will cause an overflow as 32767 is the maximum decimal value of a signed integer.

After the voltage calibration, runLength(40) is called causing a stack overflow and an endless loop.

Solution : declare *seconds* as unsigned long as well, i.e. :

```
void runLength(unsigned long seconds)
```

2) Auto calibration code in ADE9153AAPI.cpp

Changes in function: bool ADE9153AClass::StartAcal_AINormal(void)

Add = 0 to the declaration of variable ready

Move line: ready = SPI_Read_32(REG_MS_STATUS_CURRENT); inside the *while* statement.

See also bool ADE9153AClass::StartAcal_AITurbo(void) which is correct.

In function: bool ADE9153AClass::StartAcal_AV(void) Initialize variable ready to 0

3) Power and PF sign correction in ADE9153AAPI.cpp

The solution already suggested by Dlath can also be applied in the ADE9153AAOI.cpp code rather then in the Arduino code. That way the Arduino code void loop() { … can be left as it is.

In function: bool ADE9153AClass::ApplyAcal(float AICC, float AVCC)

Replace : AIGAIN = (AICC / (CAL_IRMS_CC*1000) - 1) * 134217728;

By: AIGAIN = (-(AICC / (CAL_IRMS_CC*1000)) - 1) * 134217728;