```
uint8_t scd30::readMeasurement()
{
    i2cBuff[0] = SCD30_CMMD_READ_MEAS >> 8;
    i2cBuff[1] = SCD30_CMMD_READ_MEAS & 255;
    int res = _i2c.write(SCD30_I2C_ADDR, i2cBuff, 2, false);
    if(res) return SCDnoAckERROR;

    _i2c.read(SCD30_I2C_ADDR | 1, i2cBuff, 18, false);

    uint16_t stat = (i2cBuff[0] << 8) | i2cBuff[1];
    scdSTR.co2m = stat;
    uint8_t dat = scd30::checkCrc2b(stat, i2cBuff[2]);
    if(dat == SCDcrcERROR) return SCDcrcERRORv1;

    stat = (i2cBuff[3] << 8) | i2cBuff[4];
    scdSTR.co2l = stat;
    dat = scd30::checkCrc2b(stat, i2cBuff[5]);
    if(dat == SCDcrcERROR) return SCDcrcERRORv2;

    stat = (i2cBuff[6] << 8) | i2cBuff[7];
    scdSTR.tempm = stat;
    dat = scd30::checkCrc2b(stat, i2cBuff[8]);
    if(dat == SCDcrcERROR) return SCDcrcERRORv3;

    stat = (i2cBuff[9] << 8) | i2cBuff[10];
    scdSTR.templ = stat;
    dat = scd30::checkCrc2b(stat, i2cBuff[11]);
    if(dat == SCDcrcERROR) return SCDcrcERRORv4;

    stat = (i2cBuff[12] << 8) | i2cBuff[13];
    scdSTR.humm = stat;
    dat = scd30::checkCrc2b(stat, i2cBuff[14]);
    if(dat == SCDcrcERROR) return SCDcrcERRORv5;

    stat = (i2cBuff[15] << 8) | i2cBuff[16];
    scdSTR.huml = stat;
    dat = scd30::checkCrc2b(stat, i2cBuff[17]);
    if(dat == SCDcrcERROR) return SCDcrcERRORv6;

    scdSTR.co2i = (scdSTR.co2m << 16) | scdSTR.co2l ;
    scdSTR.tempi = (scdSTR.tempm << 16) | scdSTR.templ ;
    scdSTR.humi = (scdSTR.humm << 16) | scdSTR.huml ;

    scdSTR.co2f = *(float*)&scdSTR.co2i;
    scdSTR.tempf = *(float*)&scdSTR.tempi;
    scdSTR.humf = *(float*)&scdSTR.humi;

    return SCDnoERROR;
}
```