Honeywell_SPI.h

Библиотека позволяет работать с датчиками давления Honeywell, работающих по шине SPI.

Возможности библиотеки:

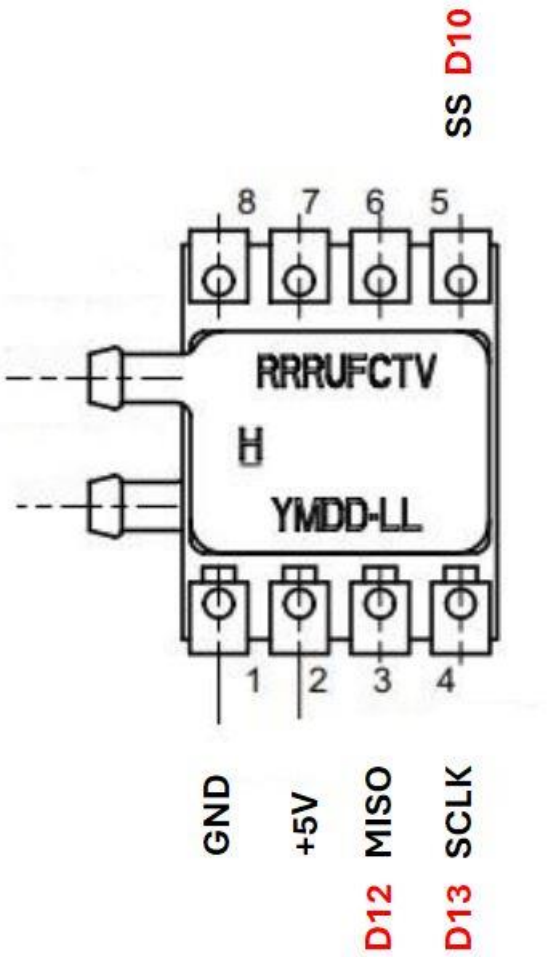
- прием давления с датчика (возможно усреднение по нескольким значениям);

- калибровка датчика;

- прием температуры с датчика (при наличии такой функции в датчике);

- подключение нескольких датчиков на одну шину.

Подключение датчика (на примере 060MDSA5):



Красным цветом отмечены пины на Arduino UNO/NANO. Для подключения двух и более датчиков, MISO и SCLK подключаются также к пинам D12 и D13 соответственно. Пины SS подключаются на любые свободные ЦИФРОВЫЕ пины (на один цифровой пин подключается один датчик- для увеличения количества используй логические микросхемы). Подключение других датчиков можно посмотреть в официальной документации, см. в папке библиотеки- «Board Mount Pressure Sensors». Сама логика работы датчика- «SPI Communication with Honeywell Digital Output Pressure».

Использование библиотеки (см. examples):

Сначала перед void setup() требуется создать объект датчика/датчиков.

Honeywell_SPI название объекта(SS_pin, pressureConst, pMin, oMin, Sum),

где SS- номер цифрового пина, куда подключен SS (максимум 255);

pMin- минимальное давление, считываемое датчиком;

oMin- см. ниже;

Sum- количество измерений в одной точке (максимум 255);

pressureConst- константа для вычисления давления, для ее вычисления требуется воспользоваться документацией «Board Mount Pressure Sensors» на странице 13. По номеру датчика нужно узнать:

- максимальное (pMax) и минимальное (pMin) давление;

- максимальное и минимальное значение с датчика (oMax, oMin)- для этого по второй букве с конца узнаем проценты от напряжения (% of Vsupply (analog)). По данным процентам вычисляем oMax и oMin. Пример: датчик **060MDSA5,** предпоследняя буква в номере «A», значит 10% to 90% of Vsupply (analog), 2^{14} counts (digital). Следовательно, oMin = 10% от 2^{14} = 1638, oMax = 90% от 2^{14} = 14745.

И в итоге:

$$\mathbf{pressureConst} = \frac{(\text{максимальное давление} - \text{минимальное давление})}{(\mathbf{oMax} - \mathbf{oMin})},$$

данное значение записывается точно максимально точно, без округления.

В void setup() требуется инициализация датчика:

название_объекта.readSensor()

Функции в void loop():

Считывание значения с датчика (данная функция только считывает значение с датчика, не возвращает в цикл сами значения температуры либо давления):

название_объекта.readSensor()

Расчет давления после readSensor():

название_объекта.getPressure()

Расчет температуры после readSensor():

название_объекта.get Temperature ()

Для получения усредненного значения давления (количество измерений в одной точке указывается при создании объекта), требуется воспользоваться следующей функцией:

название_объекта.readSensorSum()

Для уменьшения нагрузки на микроконтроллер и ускорения работы программы, можно обработку данных перенести на эксель либо на подобную программу. Для этого в библиотеке реализованы функции для получения необработанных значений давления и температуры:

название_объекта. getPressureCount()

название_объекта. get TemperatureCount()

Полученные значения (PressureCount и TemperatureCount) требуется преобразовать по следующим формулам:

$$\text{Давление} = \frac{(\text{PressureCount} - \text{oMin}) * (\text{макс. давл} - \text{мин. давл})}{(\text{oMax} - \text{oMin})} + \text{мин. давл}$$

$$\text{Температура} = \left(\frac{\text{TemperatureCount}}{2047} * 200\right) - 50$$

Заготовка файла excel есть в папке библиотеки. См. пример «load reduction».