Подпрограмма преобразования положения движка переменного резистора в код

В век повальной цифровизации мы уже привыкли к кнопкам и сенсорным экранам. При создании конструкций с использованием микроконтроллеров, иногда возникает необходимость в плавном изменении параметров кокой то величины. Будь то плавная регулировка температуры, частоты вращения двигателя, регулировка выходного напряжения БП с цифровым управлением и т.д. С точки зрения пользования такими устройствами удобно использовать обычный переменный резистор, определенное положение ползунка которого будет соответствовать определенному цифровому коду. Конечно, можно применить энкодер, клавиатуру, сенсорный экран или просто микроконтроллер с встроенным АЦП, но это сложнее и дороже. Я давно уже использую ниже описанный метод оцифровки в своих конструкциях. Вот пример схемы, где регулировка и подстройка параметров осуществляется с помощь переменных и подстроечных резисторов.



R2 и R4 – подстроечные резисторы, R10 и R12 – регулировочные. Это [схема инкубатора](https://www.kondratev-v.ru/avtomatika/inkubator-dlya-doma.html), который я сделал для себя.

Код подпрограммы получения цифрового кода с помощью переменного резистора показан на скриншоте ниже.



И так, программы получения цифрового кода на представленном скриншоте начинается с метки RC\_N и инструкцией, запрещающей все прерывания. Иначе, если обработка прерывания попадет на время выполнения команд данной подпрограммы, то полученный код будет не корректен и ошибочен. Следующая инструкция – clrf очищает регистр N - количества считываний. На строках с 74 по 76 размешены команды переключения вывода порта А – RA2, на вход. С метки Shet начинается постоянный анализ состояния данного вывода. Строка 77, инструкция btfss анализирует логический уровень на входе RA2 и до тех пор, и пока входной триггер Шмитта микроконтроллера будет выдавать логическую единицу, будет происходить инкремент регистра N. Как только на входе будет обнаружен логический ноль, то проверка входа сразу же прекращается. Это одно из условий прекращения счета проверок (получения кода). Второе условие прекращения проверок, это число, ограничивающие эти проверки. Для нашего случая это число 255 – это пример. Каждый раз, после инструкции прибавления единицы в регистр N, происходит проверка и сравнение значения числа сосчитанных проверок. За это отвечают команды, находящиеся на строках с 80 по 82. Эта ситуация возникает, если в регистре зафиксированы уже 255 проверок, а на входе все еще высокий уровень. Такое может быть, если постоянная времени RC цепочки очень большая. Данная проверка дает нам возможность получать нам нужное максимальное значение цифрового кода. И при этом нет необходимости точно подбирать емкость конденсатора. Хотя, при слишком большой его емкости некоторый сегмент окружности поворота движка переменного резистора будет не задействован.

После того, когда активная область программы попадет на метку Ready, будут выполнены тир команды переключения вывода порта А – RA2 на выход – строки 85,86 и 87. Далее выполняется команда установки вывода RA2 в единичное состояние, на выводе появляется напряжение почти равное напряжению питания микроконтроллера, происходит заряд конденсатора для последующего цикла проверок (считывания кода). Ниже представлен скриншот работы данного описываемого кода в программе Poteus.



Как работает данная подпрограмма, можно посмотреть в видеоролике. Все необходимые файлы проекта можно скачать архивом.