**Настройка работы среды Arduino для работы с контроллерами серии КМ от Sem.**

Для начала некоторые справочные данные по контроллерам КМ.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Наименование | Кол-во выводов | Тип микроконтроллера |
| 1 | КМ01-3200 | Di-3,Do-2 | ATMega168PA |
| 2 | КМ01-8600 | Di-8,Do-6 | ATMega324PA |
| 3 | КМ01-2442 | Di-2,Do-4,Ai-4,Ao-2 | ATMega324PA |
| 4 | КМ02-8600 | Di-8,Do-6 | ATMega644PA |
| 5 | КМ02-2442 | Di-2,Do-4,Ai-4,Ao-2 | ATMega1284PA |
| 6 | КМ03-8600 | Di-8,Do-6 | ATMega324PA |
| 7 | КМ01-KL10(20,40) | Ai-3,Do-2 | ATMega164PA |

Для контроллеров типа KM01-8600, KM01-2442, KM03-8600 используем одинаковую программу загрузчика.

Вначале устанавливаем среду IDE для Arduino, не буду на этом останавливаться в интернете много про это написано. Отмечу, у меня установлена v.1.8.12.

Чтобы работать с микроконтроллерами ATMega168PA, надо установить библиотеку MiniCore. Для этого заходим в Файл -> Настройки здесь выбираем «Дополнительные ссылки для Менеджера Плат» и добавляем вниз списка новую строку: **https://mcudude.github.io/MiniCore/package\_MCUdude\_MiniCore\_index.json**

Запоминаем все и теперь открываем Инструменты -> Плата:xxxxxx -> «Менеджер Плат» и в поиске набираем «**MiniCore**” и устанавливаем его (у меня версия 2.2.0).

Для работы с контроллерами ATMega164PA,ATMega324PA…, надо установить библиотеку MightyCore. Аналогично в настройках добавляем строку:

**https://mcudude.github.io/MightyCore/package\_MCUdude\_MightyCore\_index.json**

Запоминаем все и теперь открываем Инструменты -> Плата:xxxxxx -> «Менеджер Плат» и в поиске набираем «**MightyCore**” и устанавливаем его (у меня версия 2.0.8 – здесь это важно).

Теперь желательно перезапустить IDE Arduino и посмотреть, в выборе плат должно появиться два новых пункта **MiniCore** и **MightyCore** в которых можно выбрать свои наборы микроконтроллеров.

**Добавление контроллеров SemKM в пункты меню Arduino IDE.**

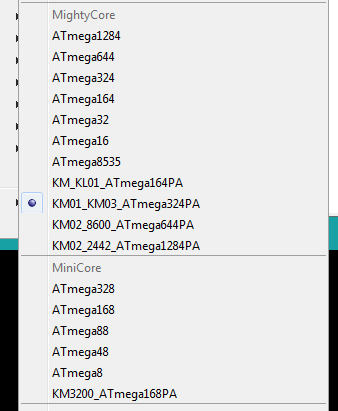
Для добавления выбора контроллеров КМ в пункты меню среды Arduino IDE, необходимо для начала определиться с месторасположением файла **board.txt** для библиотеки **MightyCore**. У меня он находиться по пути:

**C:\Users\Ed\AppData\Local\Arduino15\packages\MightyCore\hardware\avr\2.0.8**

Корректирум файл **board.txt**, предварительно сохранив старый файл в туже папку, но с другим именем (на всякий случай, например, board\_old.txt). Открыть файл с помощью, например, программы Блокнот(Notepad++), и добавить в конец файла строки из прилагаемого файла **Files\MightyCore\boardsAdd.txt**

То-же самое надо сделать с файлом **board.txt** из библиотеки **MiniCore.** Прилагаемый файл **Files\MiniCore\boardsAdd.txt**

После этих процедур перезапустить IDE Arduino и если все в порядке, найти новые пункты выбора плат в соответствующих пунктах:



Теперь можно выбирать нужный тип контроллера KM и писать для него программы.

**Загрузчик для контроллеров SemKM.**

Отдельно необходимо выделить программу загрузчика для контроллеров. Я использовал программу Optiboot, немного переделав его для загрузки через RS485. В прилагаемой папке Source находяться проекты для AtmelStudio загрузчика Optiboot, для всех контроллеров KM.

В папке Boot находятся откомпилированные файлы загрузчика, которые необходимо вставить в соответствующие папки среды Arduino.

Для библиотеки **MightyCore** вставить папку **km\_optiboot\_mighty** по пути:

**C:\Users\Ed\AppData\Local\Arduino15\packages\MightyCore\hardware\avr\2.0.8\bootloaders\optiboot\_flash\bootloaders\**

Для библиотеки **MiniCore** вставить папку **km\_optiboot\_mini** по пути:

**C:\Users\Ed\AppData\Local\Arduino15\packages\MiniCore\hardware\avr\2.2.0\bootloaders\optiboot\_flash\bootloaders**

В каждой из этих папок находятся бинарные файлы прошивок и скрипты для записи прошивки в микроконтроллер. Программирование выполняется через программатор **AVRISP mkII** драйвер которого находиться в папке Driver. При подключении программатора, он должен определяться системой без ошибок. Программирование осуществляется через разъем **ISP**.

После включения, первые 4 сек. загрузчик ждет начала прошивки программы пользователя через среду Arduino, затем переходит на основную программу, если она записана в микроконтроллер.

Советую при обновлении программы ждать конца компиляции в Arduino (полная шкала), а затем подать питание на контроллер и в течении 4 сек начнется загрузка новой программы в контроллер.

**Выводы контроллеров, соответствующие микроконтроллерам.**

Конкретные значения выводов надо смотреть по рисункам и схемам контроллеров, вот основные:

**КМ01-3200**

|  |  |  |
| --- | --- | --- |
| № | Номер вывода | Номер в Arduino |
| 1 | Di-1 | 16 |
| 2 | Di-2 | 15 |
| 3 | Di-3 | 14 |
| 4 | Do-1 | 9 |
| 5 | Do-2 | 10 |
| 6 | TXE\_RS485 | 2 |

**КМ01-8600**

|  |  |  |
| --- | --- | --- |
| № | Номер вывода | Номер в Arduino |
| 1 | Di-1 | 24 |
| 2 | Di-2 | 25 |
| 3 | Di-3 | 26 |
| 4 | Di-4 | 27 |
| 5 | Di-5 | 28 |
| 6 | Di-6 | 29 |
| 7 | Di-7 | 30 |
| 8 | Di-8 | 31 |
| 9 | Do-1 | 21 |
| 10 | Do-2 | 20 |
| 11 | Do-3 | 19 |
| 12 | Do-4 | 18 |
| 13 | Do-5 | 17 |
| 14 | Do-6 | 16 |
| 6 | TXE\_RS485 | 10 |

**КМ01-2442**

|  |  |  |
| --- | --- | --- |
| № | Номер вывода | Номер в Arduino |
| 1 | Di-1 | 26 |
| 2 | Di-2 | 27 |
| 3 | Do-1 | 15 |
| 4 | Do-2 | 14 |
| 5 | Do-3 | 16 |
| 6 | Do-4 | 17 |
| 7 | Ai-1 | 28/A4 |
| 8 | Ai-2 | 29/A5 |
| 9 | Ai-3 | 30/A6 |
| 10 | Ai-4 | 31/A7 |
| 11 | Ao-1 | 12 |
| 12 | Ao-2 | 13 |
| 13 | TXE\_RS485 | 10 |

Принципиальные схемы и соответствующие выводы микроконтроллеров в папке **Схемы КМ**.