Arduino PWM және аналогтық шығу

• Arduino-да аналогтық шығыс арналары жоқ

• Басқару сигналы үшін бізге аналог қажет (0 - 5)

• Біз екі түрлі нұсқаны қолданамыз:

- PWM-ді кернеуге айналдыратын RC Lowpass сүзгісін жасаңыз

- DAC чипін / IC қолданыңыз (сандықтан аналогтық түрлендіргішке)

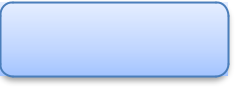
• Мұндай чипте SPI шинасы немесе I2C шинасы қолданылады

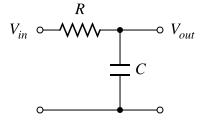
Аналогтық шығу

• Басқару сигналы үшін пайдаланылатын аналогтық шығыс жасаңыз ()

• №1 және # 2 нұсқаларын жасаңыз және тексеріңіз

• Нәтижелерді салыстырыңыз және талқылаңыз

Option 1



**1-нұсқа: PWM-ді кернеуге айналдыру**

RC Lowpass Filter:

e.g., = 3.9 Ω

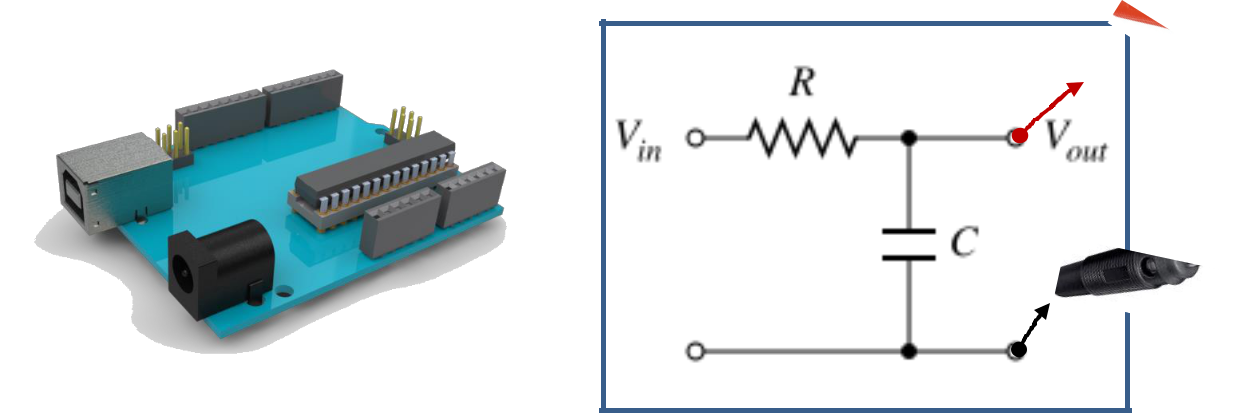
e.g., = 10

* <http://www.instructables.com/id/Analog-Output-Convert-PWM-to-Voltage/>
* [http://provideyourown.com/2011/analogwrit](http://provideyourown.com/2011/analogwrite-convert-pwm-to-voltage/) [e-convert-pwm-to-voltage/](http://provideyourown.com/2011/analogwrite-convert-pwm-to-voltage/)

**PWM-ді кернеу мысалына түрлендіру**

Төменде сіз қалай сыммен байланысуға болатынын көресіз:

|  |  |  |
| --- | --- | --- |
|  | Digital I/O Pin | Breadboard |
|  | with PWM |
| GND |  |
|  | = 3.9 Ω |

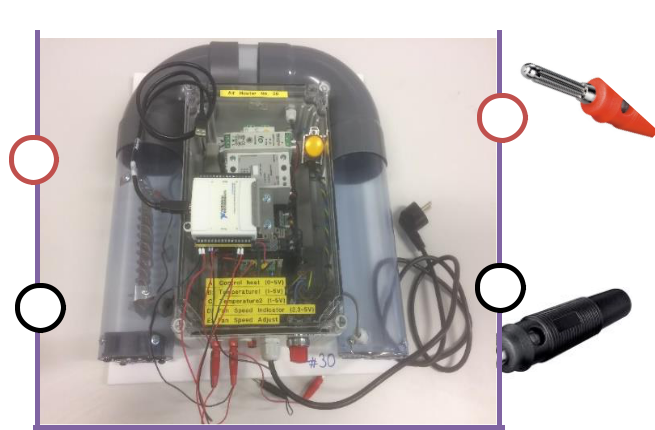


= 10

Analog In Pin

Arduino

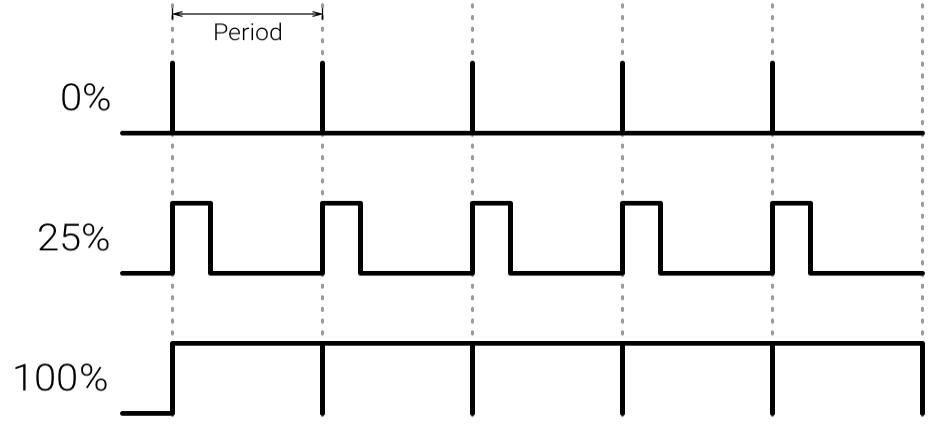
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Banana |  | Air Heater |  | Banana |
|  |  |  |
| Plugs | |  |  | Plugs |
| + | |  | + | |
|  | Process | |
| Control | |  |
|  | Value ( ) | |
| Signal ( ) | |  |
|  |  |  |



- -

Important! Test with Mathematical Model first (HIL Simulation and Testing)!

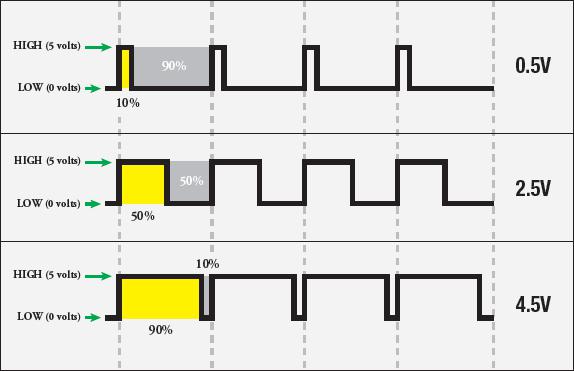
PWM



PWM - берілген жиілік пен жұмыс цикліне сәйкес тербелетін сандық (яғни квадрат толқын) сигнал. Жиілік (Гц-мен көрсетілген) шығу импульсінің қаншалықты жиі қайталанатынын сипаттайды. Период - бұл әр циклдің өтетін уақыты және оған жиіліктің кері мәні. Жұмыс циклі (пайызбен көрсетілген) сол жиілік терезесіндегі импульстің енін сипаттайды.

Сіз жұмыс циклін сигналдың орташа «қосылу» уақытын ұлғайту немесе азайту үшін реттей аласыз. Келесі диаграммада импульстік пойыздар 0%, 25% және 100% кезекшілік көрсетілген:

<https://developer.android.com/things/sdk/pio/pwm.html>



**Импульстің енін модуляциялау (PWM)**

**Импульстің енін модуляциялау (PWM)**

AnalogWrite () артындағы «таңқаларлық шындық»:

• Arduino аналогтық кернеуді (0 мен 5 вольт арасындағы) analogRead () функциясын пайдаланып оқи алатындығын білеміз.

• Arduino үшін аналогтық кернеулерді шығарудың жолы бар ма? Жауап жоқ ... және иә. Arduino-да нақты кернеудің кернеуі жоқ. Бірақ, Arduino өте жылдам болғандықтан, оны PWM деп аталатын («импульстік-кең модуляция») көмегімен қолдан жасай алады. Arduino-да «~» деген түйреуіштер PWM / Analog out үйлесімді.

• Arduino жылдамдығы соншалық, ол түйреуішті секундына 1000 рет жыпылықтай алады. PWM жыпылықтайтын пиннің ЖОҒАРЫ мен аз уақытты жұмсау уақытын өзгерту арқылы бір қадам алға жылжиды. Егер ол уақыттың көп бөлігін ЖОҒАРЫ өткізсе, сол істікшеге жалғанған жарық диодты жарық көрінеді. Егер ол көп уақытын ТӨМЕН өткізсе, жарық диоды күңгірт болып көрінеді. Штырь сіздің көзіңіз анықтағаннан гөрі жыпылықтағандықтан, Arduino «шынайы» аналогтық шығыс елесін жасайды.

• Сигналды көбірек тегістеу үшін біз RC тізбегін құрып, қолданамыз (Lowpass Filter)



**Импульстің енін модуляциялау (PWM)**

• Arduino бағдарламалау тілі PWM-ді пайдалануды жеңілдетеді; AnalogWrite қоңырау шалыңыз (pin, dutyCycle), мұндағы dutyCycle - 0-ден 255-ке дейінгі мән, ал PIN - PWM түйреуіштерінің бірі (3, 5, 6, 9, 10 немесе 11).

• analogWrite функциясы PWM аппараттық құралына қарапайым интерфейс ұсынады, бірақ жиіліктің бақылауын қамтамасыз етпейді. (Функцияның атына қарамастан, шығыс көбінесе квадрат толқын деп аталатын цифрлық сигнал екенін ескеріңіз.)0−5 →0−255 → =51



=0 → 0



= 5 → 255



= → 51∗

analogWrite():

<https://www.arduino.cc/en/Reference/AnalogWrite>

Secrets of Arduino PWM:

<https://www.arduino.cc/en/Tutorial/SecretsOfArduinoPWM>

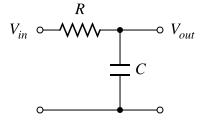
<http://techteach.no/simview/pwm_control/>

Examples: <http://techteach.no/simview/rc_circuit/>

Design and Analyze the RC Lowpass Filter based on simulations (Time domain and Frequency domain) in LabVIEW. Find proper values for R and C. Find Bandwidth/Cut-off frequency, etc.

**Дизайн және талдау**

Тереңірек түсіну үшін келесі әрекеттерді қарастыруға болады:



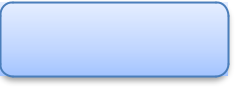
• RC Low Pass Filter үшін жіберу функциясын табыңыз.

• Дифференциалдық теңдеуді таба бастаңыз, содан кейін беру функциясын табу үшін Лапласты қолданыңыз.= =?



• LabVIEW ішінде PWM сигналын құрыңыз және имитациялаңыз.

• LabVIEW ішінде RC Lowpass Filter беру функциясын құрыңыз.

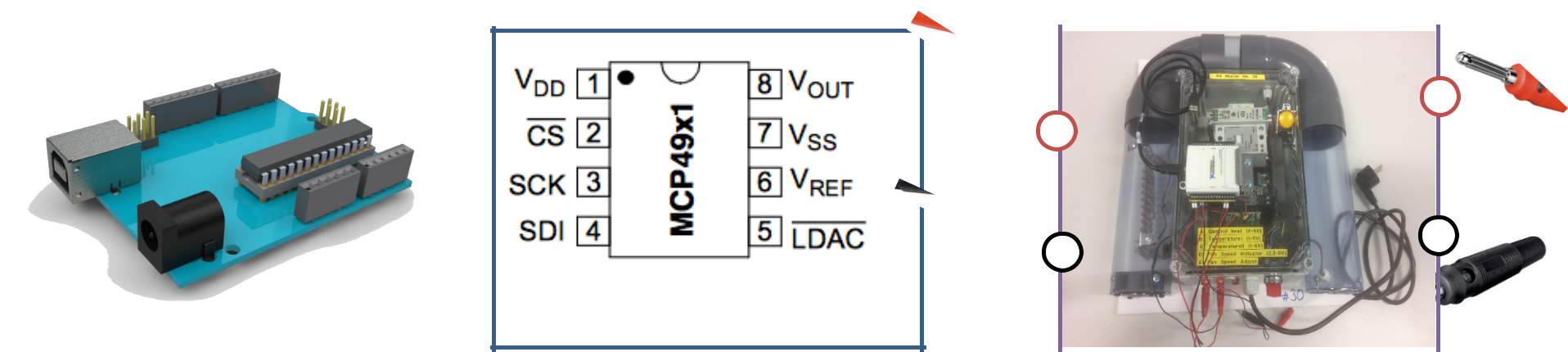
Option 2

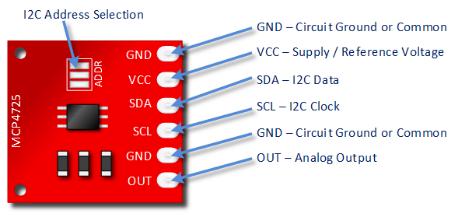


**2-нұсқа: DAC чипін қолданыңыз**

Төменде сіз қалай сыммен байланысуға болатынын көресіз:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Connect | Breadboard | Banana |  |  | Air Heater |  | Banana |
| these to |  |  |  |  |  |
|  | Plugs |  |  |  |  | Plugs |
| Arduino |  |  |  |  |  | + | |
|  |  | + | |  |  |
|  |  |  |  | Process | |
|  |  | Control | | |  |
|  |  |  | Value ( ) | |
|  |  | Signal ( ) | | |  |
|  |  |  | - | |
| Analog In Pin |  | - | | |  |
|  |  |  |  |
|  |  |  |  |  |  |  |
| Arduino |  |  |  |  |  |  |  |
|  | Important! Test with Mathematical Model | | | | | |
|  |  |
|  |  | first (HIL Simulation and Testing)! | | | |  |  |



Using a DAC chip

* DAC – Digital to Analog Converter
* Use, e.g., Microchip MCP4911, MCP4725 or similar
* **SPI** Arduino Library:<https://www.arduino.cc/en/Reference/SPI>
* MCP49XX Arduino Library: [https://github.com/exscape/electronics/tree/mas](https://github.com/exscape/electronics/tree/master/Arduino/Libraries) [ter/Arduino/Libraries](https://github.com/exscape/electronics/tree/master/Arduino/Libraries)

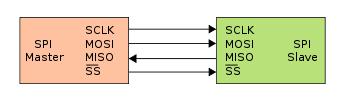
SPI Bus

• Сериялық перифериялық интерфейс (SPI) - бұл бір немесе бірнеше перифериялық құрылғылармен қысқа қашықтықта жылдам байланыс орнатуға арналған микроконтроллерлер қолданатын синхронды сериялық деректер хаттамасы.

• SPI қосылымында әрдайым перифериялық құрылғыларды басқаратын бір негізгі құрылғы болады (әдетте микроконтроллер).

• SPI құрылғылары толық дуплексті режимде бір мастермен мастер-құл архитектурасын қолдана отырып байланысады.

• Интерфейсті Motorola әзірледі және іс жүзінде стандартқа айналды.

• Әдеттегі қосымшаларға датчиктер, Secure Digital карталары және сұйық кристалды дисплейлер (LCD) кіреді.

SCLK : Serial Clock (output from master)

MOSI : Master Output, Slave Input (output from master)

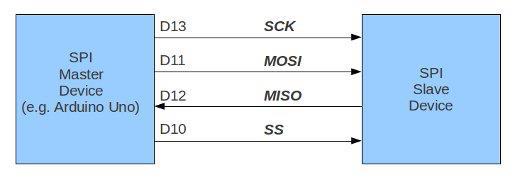
MISO : Master Input, Slave Output (output from slave)

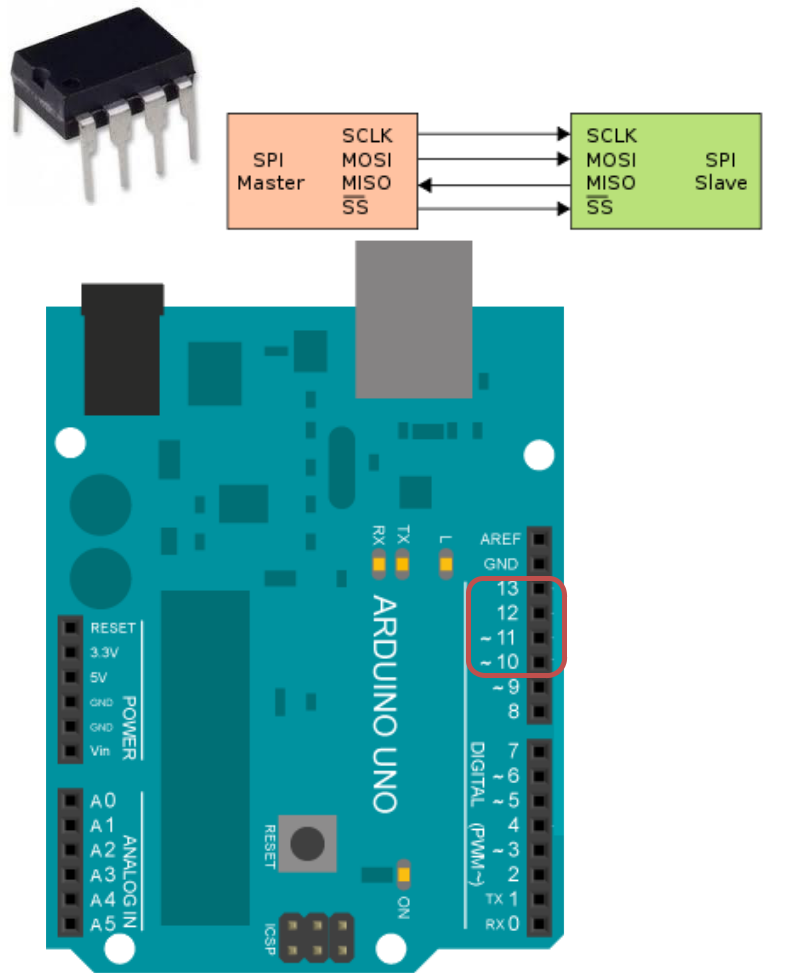
|  |  |
| --- | --- |
| SS (or SC) : Slave Select (active low, output from master) | <http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus> |
|  |

<https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi>

Arduino SPI

* <https://www.arduino.cc/en/Reference/SPI>
* <http://tronixstuff.com/2011/05/13/tutorial-arduino-and-the-spi-bus/>
* [http://arduino.stackexchange.com/questions/163](http://arduino.stackexchange.com/questions/16348/how-do-you-use-spi-on-an-arduino) [48/how-do-you-use-spi-on-an-arduino](http://arduino.stackexchange.com/questions/16348/how-do-you-use-spi-on-an-arduino)
* <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi>





|  |  |  |
| --- | --- | --- |
| MCP4911: 10-bit single DAC | | DAC |
| Arduino | MCP4911 |

= 5

= 0

SCK (13)

MISO (12)

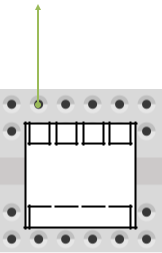
MOSI (11)

SS (10)

MISO Not Used, since we get nothing back from DAC IC

LDAC кірісі құрылғыны таңдау үшін пайдаланылуы мүмкін, және сіз құрылғыны осы істікшемен қосу және өшіру үшін GPIO штифтін пайдалануға болады. Бұл мысалда біз оны тек жерге байлап қоямыз, сонда ол әрқашан таңдалады және жұмыс істейді.

Analog Out (0-5V)



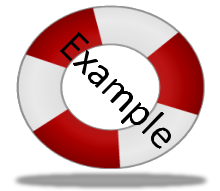
|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |

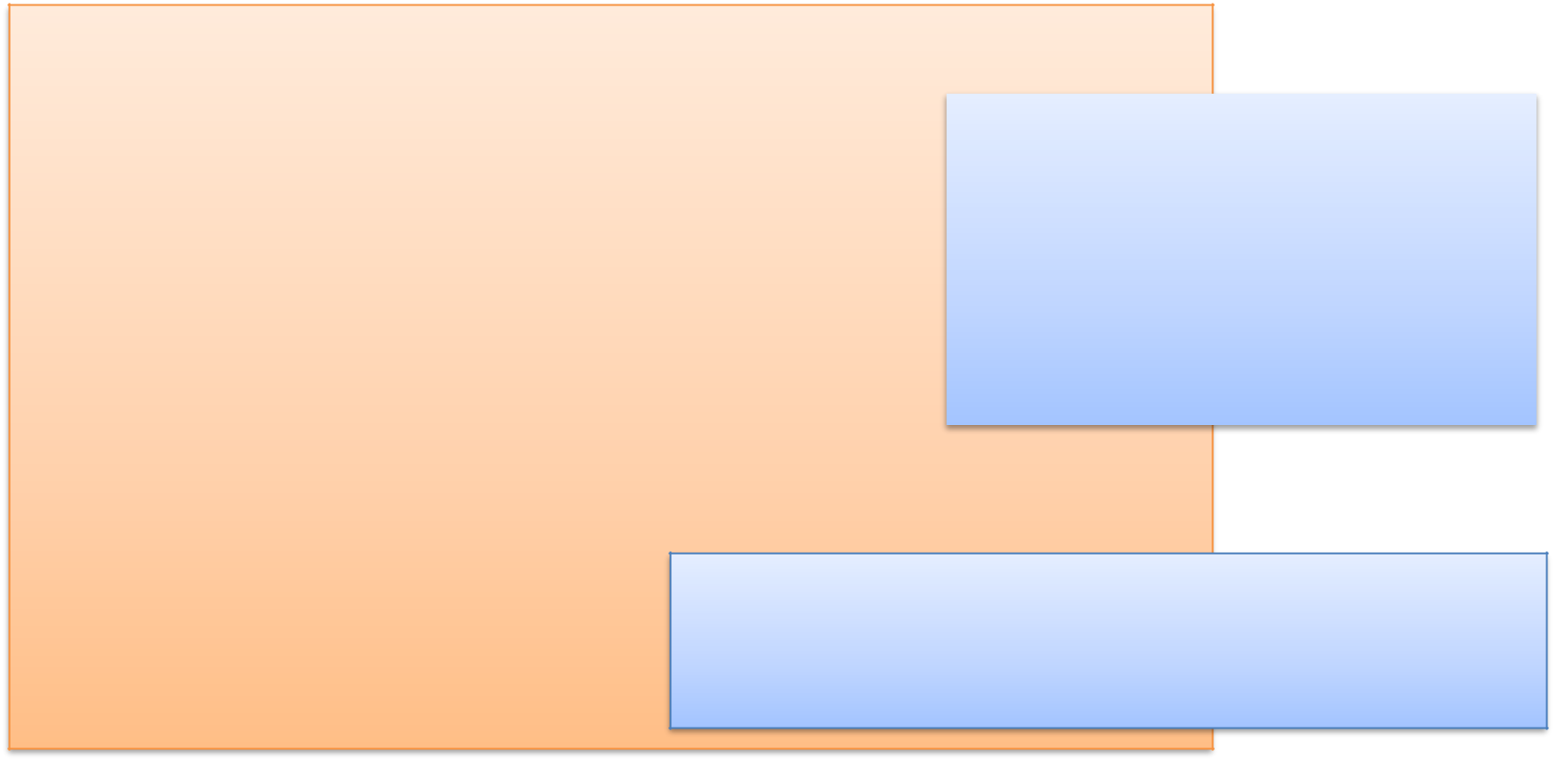
|  |  |  |  |
| --- | --- | --- | --- |
| 8 | 7 | 6 | 5 |

 MCP4911

1  2  3  4

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

MCP49xx Arduino Library Example



|  |  |  |  |
| --- | --- | --- | --- |
| **#include <SPI.h>** | //Include the Arduino SPI Library | |  |
| **#include <DAC\_MCP49xx.h>** //Include the MCP49xx Arduino Library | | |  |
|  | |  |  |
| // The Arduino pin used for the slave select / chip select | |  | The control signal (u) should |
| **#define SS\_PIN 10** |  |  |
| //Set up the DAC DAC MCP4911 | |  | come from the PI/PID |
|  | controller function. |
| **DAC\_MCP49xx dac(DAC\_MCP49xx::MCP4911, SS\_PIN);** | |  |
|  |  |
| void setup() |  |  | It need to be converted from 0- |
| { |  |  | 5V (or 0-100%) -> 0-1023 |
| } |  |  |
| void loop() |  |  | before we send it to the DAC |
| { |  |  |  |
|  |  |  |

double u; //Control Signal

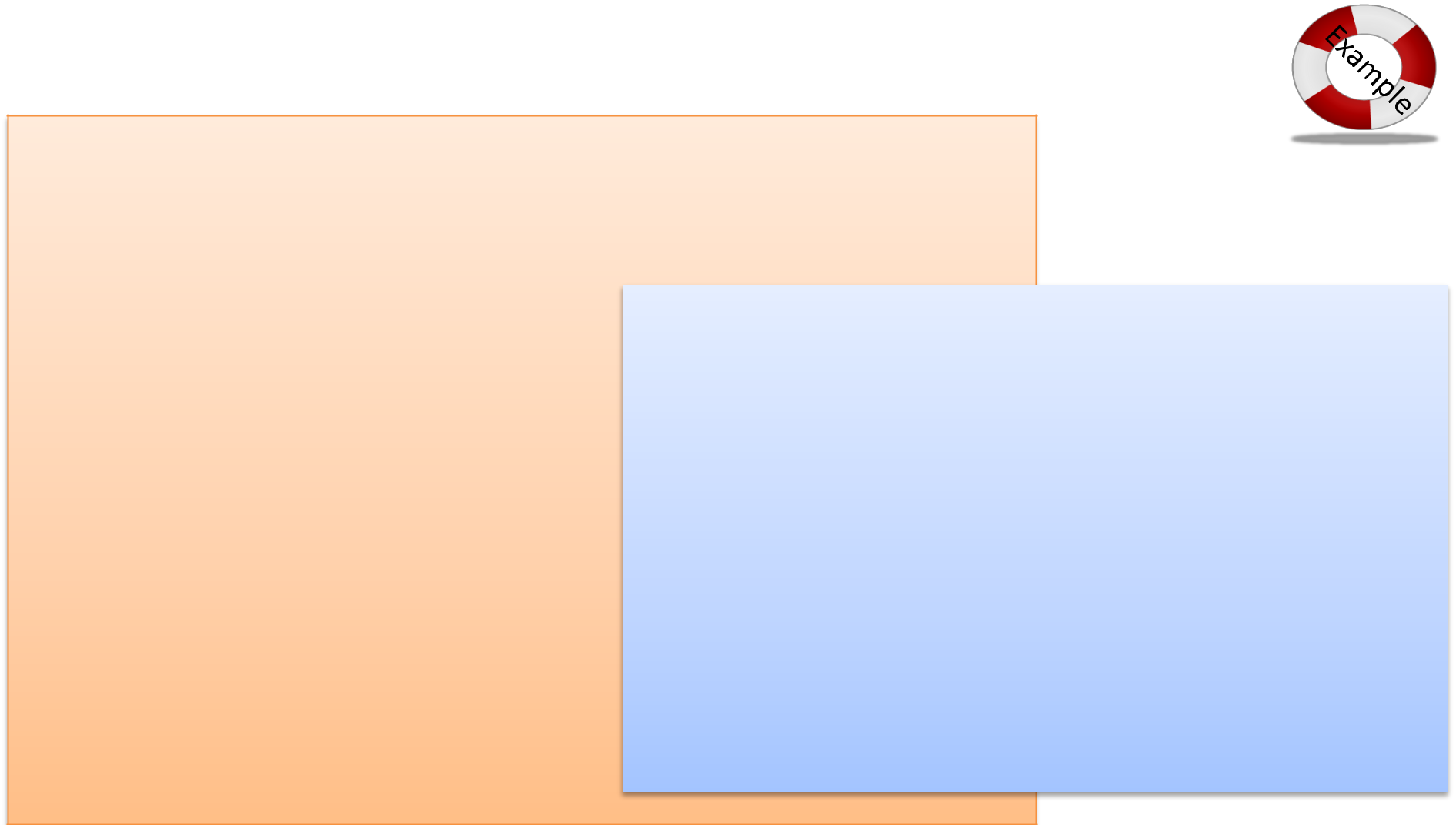
* For MCP4911, use values below (but including) 1023 (10 bit) u = 255; //Simulating the Control Value

**dac.output(u);**

|  |  |
| --- | --- |
| delay(5000); | Connect the circuit (Arduino + MCP4911) on a |
|  |
| u = 512; //Simulating the Control Value | breadboard. Use a multi-meter so see if you get |
| dac.output(u); |
| the correct output signal |
| delay(5000); |
|  |

}

|  |
| --- |
| <https://github.com/exscape/electronics/tree/master/Arduino/Libraries> |



MCP49xx Arduino Library Example

#include <SPI.h> #include <DAC\_MCP49xx.h>

//Include the Arduino SPI Library

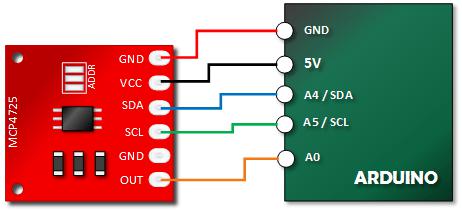
//Include the MCP49xx Arduino Library

* The Arduino pin used for the slave select / chip select #define SS\_PIN 10

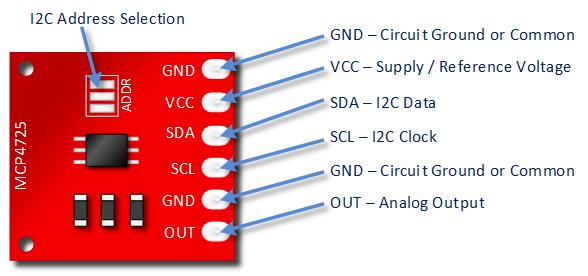
DAC\_MCP49xx dac(DAC\_MCP49xx::MCP4911, SS\_PIN);

|  |  |
| --- | --- |
| void setup() | Connect the circuit (Arduino + MCP4911) on a |
|  |
| { | breadboard. Use a multi-meter so see if you get the |
| Serial.begin(9600); |
|  |
| } |  |
| void loop() | correct output signal. |
|  |
| { |  |
| double u; //Control Signal | On the Multimeter you should see the output slowly |
| int aiPin = 0; |
|  |
| int aiValue; |  |
| for (int i=0; i<1023; i++) | increasing from ~0V to ~5V with intervals of 1000ms. |
|  |
| { |  |
| u = i; | You can also connect the output from the DAC to an |
| dac.output(u); |
| Analog Input Pin on the Arduino. Write the value to |
| aiValue = analogRead(aiPin); |
|  |
| Serial.print("AIValue="); | the Serial Monitor. |
| Serial.println(aiValue); |
|  |
| delay(1000); |  |
| } |  |
| } |  |
|  |

Alternative Solution



MCP4725



|  |  |
| --- | --- |
| 12-bit resolution |  |
| I2C Interface | The MCP4725 is a little more expensive |
|  | (than MCP49xx), but simpler to use. |