AN3636

Using the Internal OPAMP as Regulated Power Supply for MVIO

Introduction

Author: Johan Vaarlid, Microchip Technology Inc.

This application note will explain how and when to use the analog signal condition(OPAMP) peripheral available on the Microchip AVR® DB Family of microcontrollers as an adjustable power supply. The internal op amps have internal resistor ladders capable of outputting eight derivatives of VDD completely without external components.

In a multitude of applications, it is necessary to use two or more voltage domains to fulfill the design specifications, either for power saving reasons or due to specific components needing a specific I/O level. This has traditionally led to the necessity of extra power supplies.

This power supply feature will be showcased by acting as the second power supply for the Multi-Voltage I/O (MVIO). The MVIO peripheral allows for a subset of ports in the AVR® to operate on a separate voltage domain VDDIO2. The voltage range of VDDIO2 is the same as for VDD, although the Brown-out Detector (BOD) for VDDIO2 is fixed and has a typical value of 1.6V.

Utilizing the internal ADC, it is possible to verify the voltage level on the output of the op amp. This can be useful for functional safety applications where it is necessary to have a stable and/or accurate voltage level before/under use to ensure proper operation.

The example code for replicating the results described in this application note is available on GitHub.

Additional details on OPAMP performance and general configuration are available in the device data sheet.

Features

- Eliminates the Need for Extra External Power Supply for Second Voltage Domain
- The Regulated Power Supply can Output Voltages Between V_{DD}-1V and 0V
- Up to 20 mA Current
- Internal Resistor Ladder to Support Eight Derivatives of V_{DD} Without External Components
- Run-time Adjustable
- Ability to Shut Down the Second Voltage Domain Completely

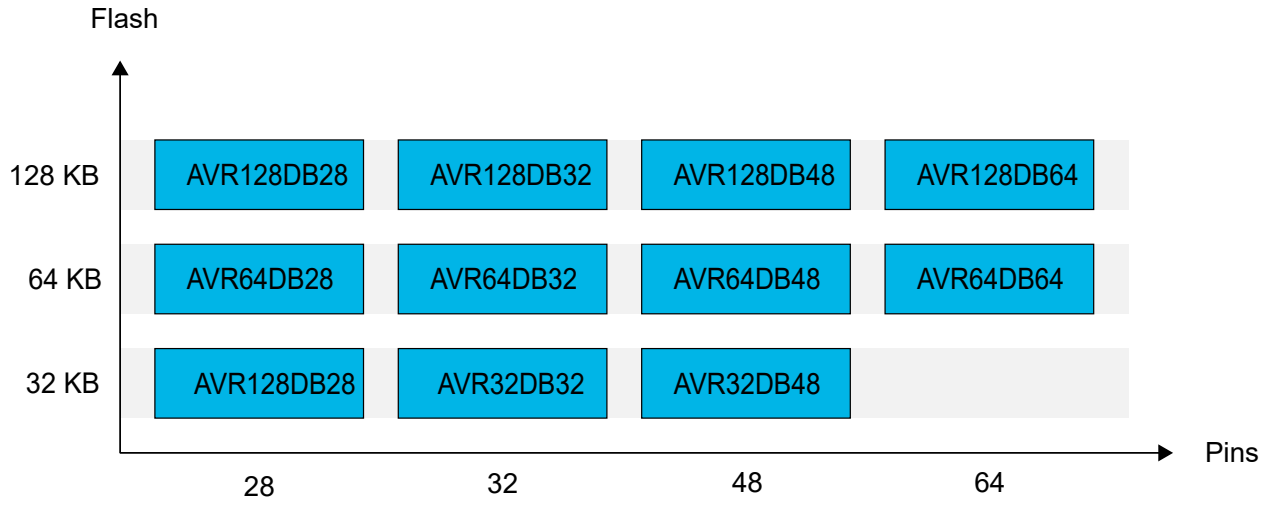
Table of Contents

1. Relevant Devices

This section lists the relevant devices for this document. The following figures show the different family devices, laying out pin count variants and memory sizes:

- Vertical migration upwards is possible without code modification, as these devices are pin-compatible and provide the same or more features
- Horizontal migration to the left reduces the pin count and, therefore, the available features
- Devices with different Flash memory sizes typically also have different SRAM and EEPROM

Figure 1-1. AVR® DB Family Overview



2. Overview

The Analog Signal Conditioning (OPAMP) peripheral features one, two or three operational amplifiers (op amps), designated OPn where n is zero, one or two. These op amps are implemented with a flexible connection scheme using analog multiplexers and resistor ladders. This allows a large number of analog signal conditioning configurations to be achieved, many of which require no external components. A multiplexer at the non-inverting (+) input of each op amp allows connection to either an external pin, a wiper position from a resistor ladder, a DAC output, ground, or $V_{DD}/2$. A second multiplexer at the inverting (-) input of each op amp allows connection to either an external pin, a wiper position from a resistor ladder, the output of the op amp, or DAC output. Three more multiplexers connected to each resistor ladder provide additional configuration flexibility. Two of these multiplexers select the top and bottom connections to the resistor ladder, and the third controls the wiper position.

Figure 2-1. Block Diagram

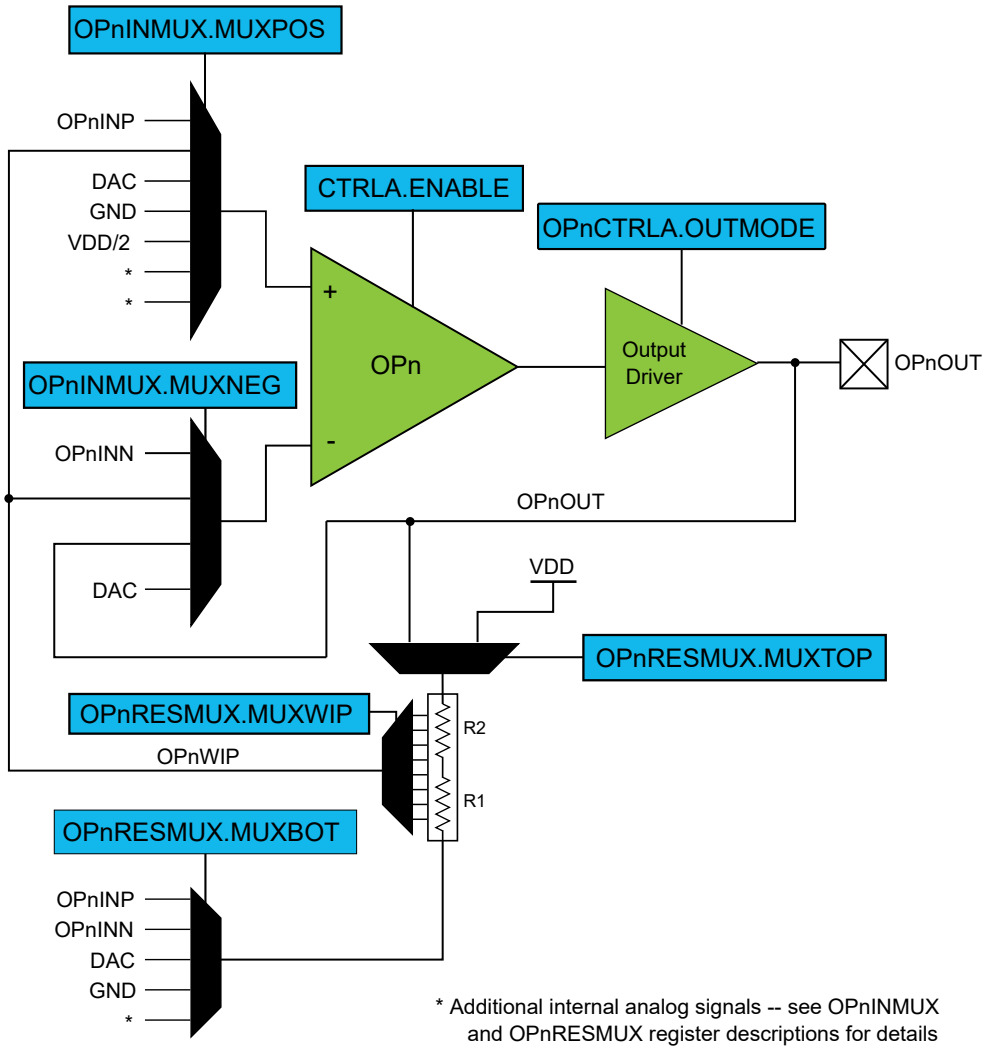


Table 2-1. Signal Description

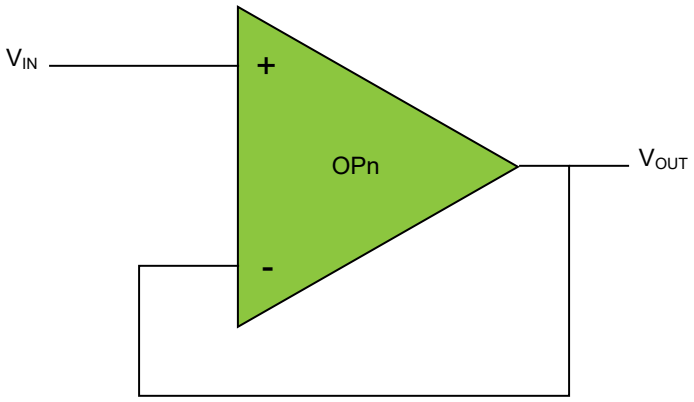
Signal Name	Type	Description
OPnINP	Analog input	Non-inverting (+) input pin for OPn

.....continued		
Signal Name	**Type**	**Description**
OPnINN	Analog input	Inverting (-) input pin for OPn
OPnOUT	Analog output	Output from OPn

2.1 Voltage Follower

The OPAMP peripheral is highly flexible and can be used in various analog signal conditioning applications. One of these applications is the voltage follower.

Figure 2-2. Voltage Follower



The figure above displays a voltage follower, also known as a unity-gain buffer. The non-inverting (+) input is connected to a pin, and the output is connected to the inverting (-) input. The multiplexer settings required for this configuration are:

Table 2-2. Voltage Follower Input Configurations

	MUXPOS	MUXNEG
OP0	Voltage reference	OUT

3. Hardware Configuration

3.1 Using the AVR128DB48 Curiosity Nano

To illustrate the capabilities of the internal op amps, an AVR128DB48 Curiosity Nano is used with V_{DD} set to 5V. This can be done through Atmel Studio, MPLAB® X, or by dragging and dropping a .txt file with only the string “CMD:5V0” onto the “CURIOSITY” mass storage device that appears when the kit is connected to a PC.

To enable MVIO powered by the internal op amps on this development board, perform the below steps:

1. Remove the 0-ohm resistor right below the VDDIO2 label to cut the connection between VDD and VDDIO2.
2. Connect a wire between PD2 and VDDIO2 +, as shown below.

Refer to the AVR128DB48 Curiosity Nano User Guide for information on how to reverse this process.

Figure 3-1. Curiosity Nano AVR128DB48

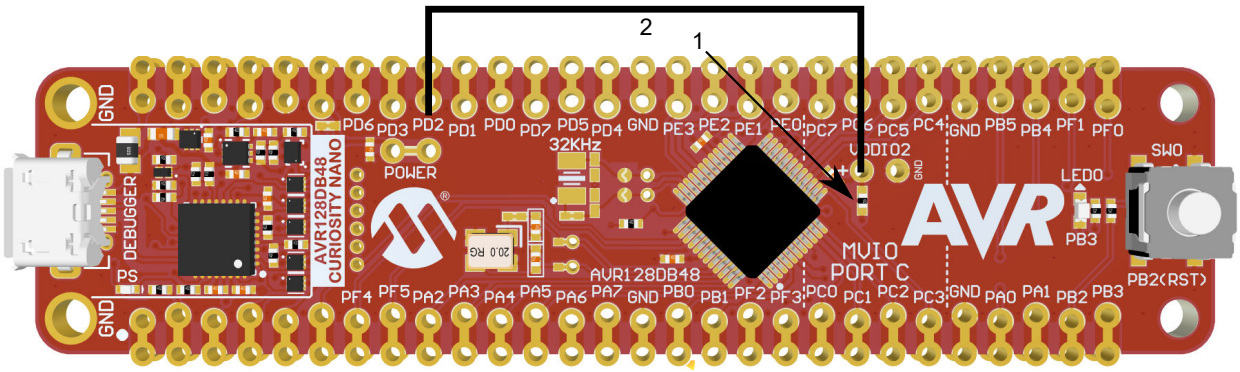
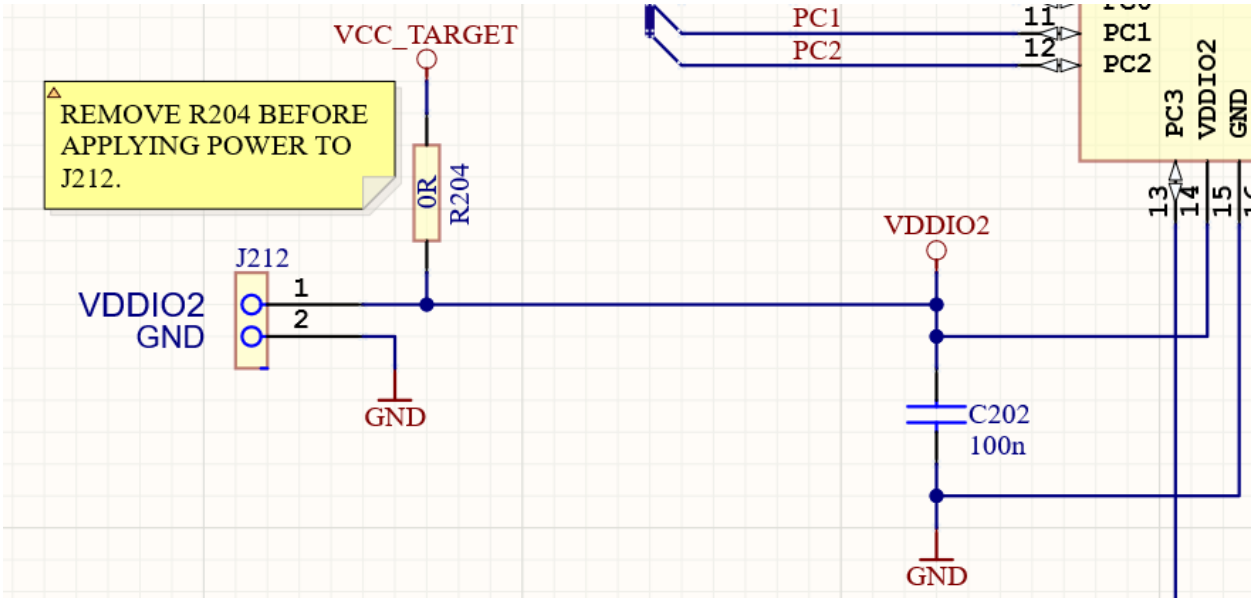


Figure 3-2. Removing of 0-Ohm Resistor to Cut the Connection Between VDDIO2 and VDD



3.2 Supplying Power To VDDIO2

The Multi-Voltage I/O (MVIO) can be configured in one of two supply modes by writing to the MVSYSCFG bit field in the SYSCFG1 fuse:

- Single-Supply mode, where the MVIO-capable I/O pins are powered at the same voltage level as the non-MVIO capable pins, i.e., VDD. The user must connect the VDDIO2 pin(s) to the VDD pin(s).
- Dual-Supply mode, where the MVIO-capable I/O pins are supplied by the VDDIO2 voltage, which may be different from the voltage supplied to the VDD pin(s).

Figure 3-3. FUSE.SYSCFG1 - Enabling Dual Supply Mode

8.8.2.5 System Configuration 1

Name: SYSCFG1
Offset: 0x06
Default: 0x08
Property: -

The default value given in this fuse description is the factory-programmed value, and should not be mistaken for the Reset value.

Bit	7	6	5	4	3	2	1	0
				MVSYSCFG[1:0]		SUT[2:0]		
Access				R	R	R	R	R
Default				0	1	0	0	0

Bits 4:3 – MVSYSCFG[1:0]
This bit field controls the power supply mode.

Value	Name	Description
0x00	Reserved	-
0x01	DUAL	Device used in a dual supply configuration.
0x02	SINGLE	Device used in a single supply configuration.
0x03	Reserved	-

4. OPAMP Initialization

For all the following examples, the OPAMP will use a basic initialization preparing it for use. The following steps are needed to initialize the OPAMP:

1. Disable the digital input buffer of the output pin of the OPAMP to reduce noise and power consumption.
2. Set the TIMEBASE register to be ≥ 1 µs.
3. Set the ALWAYSON bit to enable the OPAMP to run continuously.
4. Set the OUTMODE to NORMAL, to connect the I/O of the OPAMP to their respective pins.
5. Activate OPAMP in standby sleep (optional).
6. Enable the OPAMP by writing a '1' to the ENABLE bit.

The above steps, including optional steps, are implemented in the function below. This function will be used in all the following examples.

```
#define OPAMP_MAX_SETTLE_TIME 0x7F

void op_amp_init()
{
    /*Disable input on op amp output pin*/
    PORTD.PIN2CTRL = PORT_ISC_INPUT_DISABLE_gc;

    /*Set up op amp*/
    OPAMP.CTRLA = OPAMP_ENABLE_bm;
    OPAMP.TIMEBASE = (uint8_t) ceil(CLK_PER*0.000001)-1; /*Number of peripheral clock cycles
that amounts to 1us*/
    OPAMP.OP0CTRLA = OPAMP_RUNSTBY_bm | OPAMP_ALWAYSON_bm | OPAMP_OP0CTRLA_OUTMODE_NORMAL_gc;
       OPAMP.OP0SETTLE = OPAMP_MAX_SETTLE_TIME; //As the settle time is unknown, the maximums
should be set
}
```

5. Generating Voltage reference

Make sure that the steps in the 3. Hardware Configuration section is followed before continuing to implement the following examples.

5.1 Using the Internal Resistor Ladder

The internal resistor ladder has several connection points accessible with a wiper, which can be selected by writing to the MUXWIP bit field in the OPnRESMUX register. The voltage level at the wiper position will be a function of V_{in}, R_1 and R_2.

$$V_{Out} = \frac{R_1}{R_1 + R_2} V_{in}$$

Table 5-1. Resistor Ladder Outputs with VDD=5V

MUXWIP Setting	Fraction of VDD	Approximate Output Voltage
0x2	3/4	3.75V
0x3	1/2	2.5V
0x4	3/8	1.88V

To enable the internal resistor ladders, and make the OPAMP into a power supply, the following steps are needed:

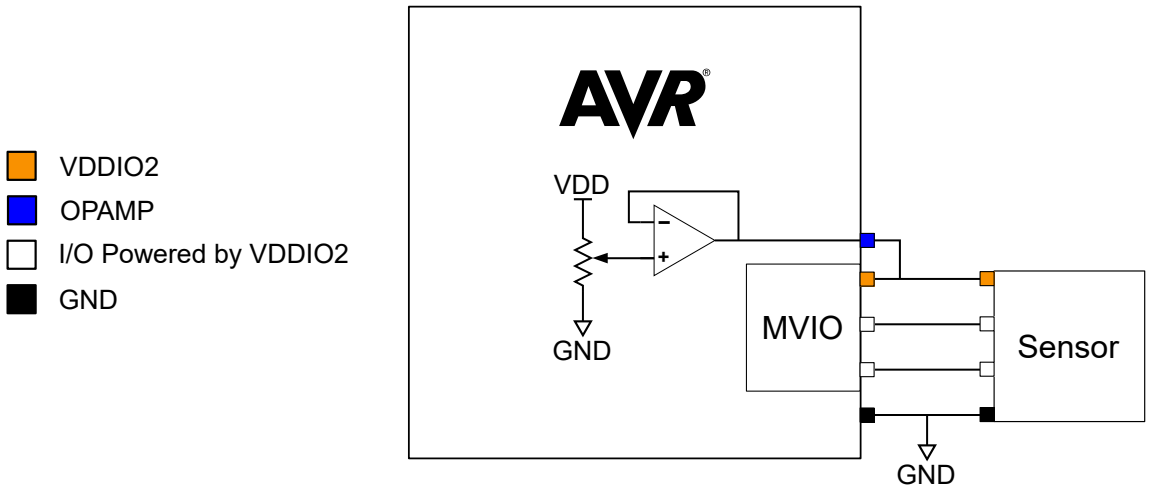
1. Connect the negative input of the OPAMP to its output.
2. Connect the positive input of the OPAMP to the wiper of the resistor ladder.
3. Connect the top of the resistor ladder to VDD.
4. connect the bottom of the resistor ladder to GND.
5. Set the wiper position to the desired position.

The function below takes a wiper position as input defined by the data sheet and sets up the OPAMP to act as a power supply.

```
void op_amp_setup_int_resistors(OPAMP_OP0RESMUX_MUXWIP_t MUXWIP_gc)
{
    OPAMP.OP0INMUX = OPAMP_OP0INMUX_MUXNEG_OUT_gc | OPAMP_OP0INMUX_MUXPOS_WIP_gc;
    OPAMP.OP0RESMUX = OPAMP_OP0RESMUX_MUXTOP_VDD_gc|OPAMP_OP0RESMUX_MUXBOT_GND_gc|MUXWIP_gc;
}
```

When using this method to supply power to the MVIO, and considering the electrical limitations of this power supply implementation, only the three wiper positions listed in table Resistor ladder outputs with VDD=5V gives an output that is useful for this application.

Figure 5-1. Using the Internal Resistor Ladder



The code below will enable the functionality mentioned above:

```
#define F_CPU 4000000ul
#define CLK_PER 4000000ul
#define OPAMP_MAX_SETTLE_TIME 0x7F
#include <avr/io.h>
#include <math.h>

void op_amp_init();
void op_amp_setup_int_resistors(uint8_t MUXWIP_gc);

int main(void)
{

    op_amp_init();
    op_amp_setup_int_resistors(OPAMP_OP0RESMUX_MUXWIP_WIP4_gc);
    while (1)
    {
        /*Your application goes here*/
    }
}


void op_amp_init()
{
    /*Disable input on op amp output pin*/
    PORTD.PIN2CTRL = PORT_ISC_INPUT_DISABLE_gc;

    /*Set up op amp*/
    OPAMP.CTRLA = OPAMP_ENABLE_bm;
    OPAMP.TIMEBASE = (uint8_t) ceil(CLK_PER*0.000001)-1; /*Number of peripheral clock cycles
that amounts to 1us*/
    OPAMP.OP0CTRLA = OPAMP_RUNSTBY_bm | OPAMP_ALWAYSON_bm | OPAMP_OP0CTRLA_OUTMODE_NORMAL_gc;
    OPAMP.OP0SETTLE = OPAMP_MAX_SETTLE_TIME; //As the settle time is unknown, the maximums
should be set
}

void op_amp_setup_int_resistors(OPAMP_OP0RESMUX_MUXWIP_t MUXWIP_gc)
{
    OPAMP.OP0INMUX = OPAMP_OP0INMUX_MUXNEG_OUT_gc | OPAMP_OP0INMUX_MUXPOS_WIP_gc;
    /* MUXWIP is the bit field that decides the output voltage of the op amp
       With VDD at 5V you will get:
       OPAMP_OP0RESMUX_MUXWIP_WIP4_gc -> ~1.88V
       OPAMP_OP0RESMUX_MUXWIP_WIP3_gc -> ~2.5V
       OPAMP_OP0RESMUX_MUXWIP_WIP2_gc -> ~3.75V */
    OPAMP.OP0RESMUX = OPAMP_OP0RESMUX_MUXTOP_VDD_gc|OPAMP_OP0RESMUX_MUXBOT_GND_gc|MUXWIP_gc;
}
```

The code for this example is available in the **using-internal-resistor-ladder** folder in these github repositories:

View Code Example on GitHub
Click to browse repository

5.2 Using External Resistors

By utilizing external resistors, virtually any voltage level between V_{DD} - 1V and 1.8V can be supplied to the positive input of the op amp. By connecting resistors as shown in the block diagram below, the following formula will give the voltage output of the voltage following

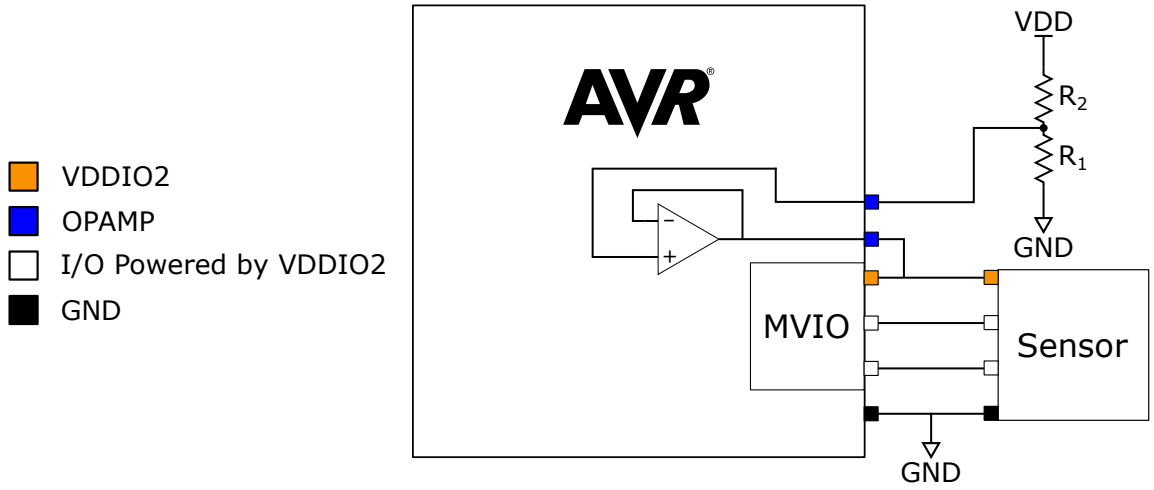
$$V_{out} = \frac{R_1}{R_1 + R_1} VDD$$

To connect the OPAMP to the external resistors, it is necessary to connect the positive input of the OPAMP to its respective pin. Also, the negative input of the OPAMP needs to be connected to its output.

The following function will perform the necessary setup:

```
void op_amp_setup_ext_resistors()
{
    OPAMP.OP0INMUX = OPAMP_OP0INMUX_MUXNEG_OUT_gc | OPAMP_OP0INMUX_MUXPOS_INP_gc;
}
```

Figure 5-2. Using an External Resistor Ladder



The code below will set up the OPAMP to output the voltage supplied to the OPnINP pin.

```
#define F_CPU 4000000ul
#define CLK_PER 4000000ul
#define OPAMP_MAX_SETTLE_TIME 0x7F
#include <avr/io.h>
#include <math.h>


void op_amp_setup_ext_resistors();
void op_amp_init();

int main(void)
{
    op_amp_init();
    op_amp_setup_ext_resistors();
    while (1)
    {
        /*Your application goes here*/
```

```
    }
}

void op_amp_init()
{
    /*Disable input on op amp output pin*/
    PORTD.PIN2CTRL = PORT_ISC_INPUT_DISABLE_gc;

    /*Set up op amp*/
    OPAMP.CTRLA = OPAMP_ENABLE_bm;
    OPAMP.TIMEBASE = (uint8_t) ceil(CLK_PER*0.000001)-1; /*Number of peripheral clock cycles
that amounts to 1us*/
    OPAMP.OP1CTRLA = OPAMP_RUNSTBY_bm | OPAMP_ALWAYSON_bm | OPAMP_OP1CTRLA_OUTMODE_NORMAL_gc;
    OPAMP.OP0SETTLE = OPAMP_MAX_SETTLE_TIME; //As the settle time is unknown, the maximums
should be set
}

void op_amp_setup_ext_resistors()
{
    OPAMP.OP0INMUX = OPAMP_OP0INMUX_MUXNEG_OUT_gc | OPAMP_OP0INMUX_MUXPOS_INP_gc;
}
```

The code for this example is available in the **using-external-resistors** folder in these github repositories:

View Code Example on GitHub
Click to browse repository

5.3 Using the Internal DAC

Combining the advantages of both the previous methods, we can use the internal Digital to Analog Converter (DAC). No external components are needed, and one can provide virtually any voltage to the positive input of the voltage follower.

Setting the DAC reference to VDD, the DATA register value for the desired output voltage becomes:

$$DATA = \frac{VDD}{1023} V_{OUT}$$

The following helper function will set the output voltage of the DAC to be fed into the OPAMP.

```
void dac_set_voltage(float v_out)
{
    /*Ensure we set the output voltage within the limits of the OPAMP drive capabilities*/
    if (v_out>VDD-1)
    {
        v_out=VDD-1;
    }
    volatile uint16_t data = (v_out/VDD)*1023;
    DAC0.DATA = data << DAC_DATA_gp;
}
```

A basic setup function for the DAC is useful. For this application, it is necessary to do the following:

1. Set the reference for the DAC to VDD.
2. Enable both the DAC and the DAC output.

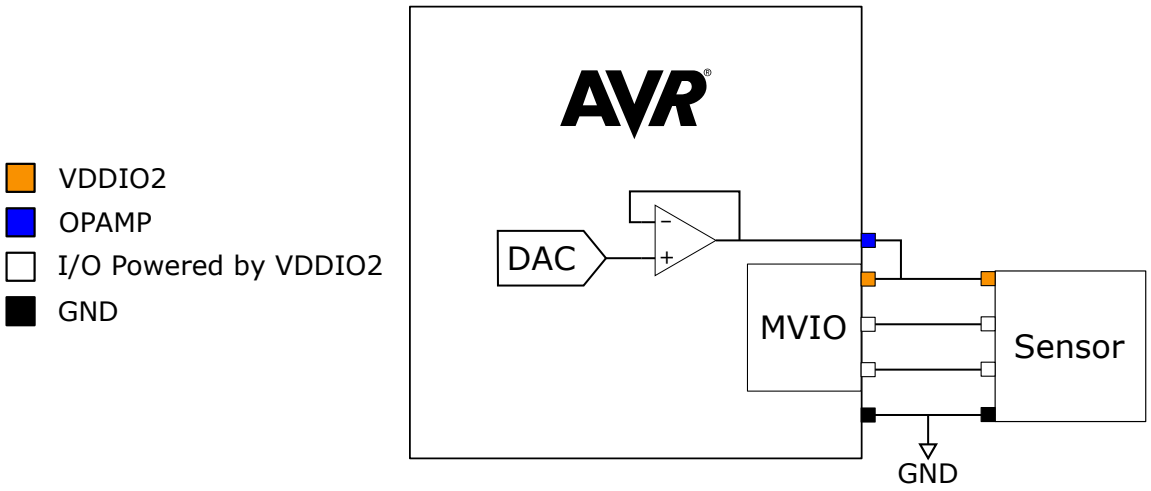
```
void dac_init()
{
    /*Setting the reference for the DAC to VDD*/
    VREF.DAC0REF = VREF_REFSEL_VDD_gc;

    /*Enable DAC and enable the DAC output*/
    DAC0.CTRLA = DAC_ENABLE_bm | DAC_OUTEN_bm;
}
```

To connect the DAC output to the positive input of the OPAMP, the following setup function is made:

```
void op_amp_setup_dac()
{
    OPAMP.OP0INMUX = OPAMP_OP0INMUX_MUXNEG_OUT_gc | OPAMP_OP0INMUX_MUXPOS_DAC_gc;
}
```

Figure 5-3. Using Internal DAC



The code below provides the above functionality.

```
#define F_CPU 4000000ul
#define CLK_PER 4000000ul
#define OPAMP_MAX_SETTLE_TIME 0x7F
#include <avr/io.h>
#include <math.h>

#define VDD 5

void op_amp_init();
void op_amp_setup_dac();
void dac_init();
void dac_set_voltage(float v_out);

int main(void)
{
    dac_init();
    op_amp_init();
    /*Default voltage is set to 1.8V*/
    dac_set_voltage(1.8);
    while (1)
    {
        /*Your application goes here*/
    }
}

void dac_set_voltage(float v_out)
{
    /*Ensure we set the output voltage within the limits of the OPAMP drive capabilities*/
    if (v_out>VDD-1)
    {
        v_out=VDD-1;
    }
    volatile uint16_t data = (v_out/VDD)*1023;
    DAC0.DATA = data << DAC_DATA_gp;
}

void dac_init()
{
    /*Setting the reference for the DAC to VDD*/
    VREF.DAC0REF = VREF_REFSEL_VDD_gc;
```

```
    /*Enable DAC and enable the DAC output*/
    DAC0.CTRLA = DAC_ENABLE_bm | DAC_OUTEN_bm;
}

void op_amp_init()
{
    /*Disable input on op amp output pin*/
    PORTD.PIN2CTRL = PORT_ISC_INPUT_DISABLE_gc;

    /*Set up op amp*/
    OPAMP.CTRLA = OPAMP_ENABLE_bm;
    OPAMP.TIMEBASE = (uint8_t) ceil(CLK_PER*0.000001)-1; /*Number of peripheral clock cycles
that amounts to 1us*/
    OPAMP.OP0CTRLA = OPAMP_RUNSTBY_bm | OPAMP_ALWAYSON_bm | OPAMP_OP0CTRLA_OUTMODE_NORMAL_gc;
    OPAMP.OP0SETTLE = OPAMP_MAX_SETTLE_TIME; //As the settle time is unknown, the maximums
should be set
}
void op_amp_setup_dac()
{
    OPAMP.OP0INMUX = OPAMP_OP0INMUX_MUXNEG_OUT_gc | OPAMP_OP0INMUX_MUXPOS_DAC_gc;
}
```

The code for this example is available in the **using-internal-DAC** folder in these github repositories:

6. Monitoring the Voltage Level with the Internal ADC

The Internal Analog to Digital Converter (ADC) makes it possible to monitor and measure the output voltage level of the voltage follower. To do this, it is necessary to connect the output pin of the voltage follower to the ADC, as shown in Figure 6-1.

The ADC is initialized to perform this functionality with the following steps:

1. Set the voltage reference of the ADC to 4.096V.
2. Set the input of the ADC to the output pin of the OPAMP.
3. Set some oversampling to get a more accurate and averaged result.
4. Enable the ADC.

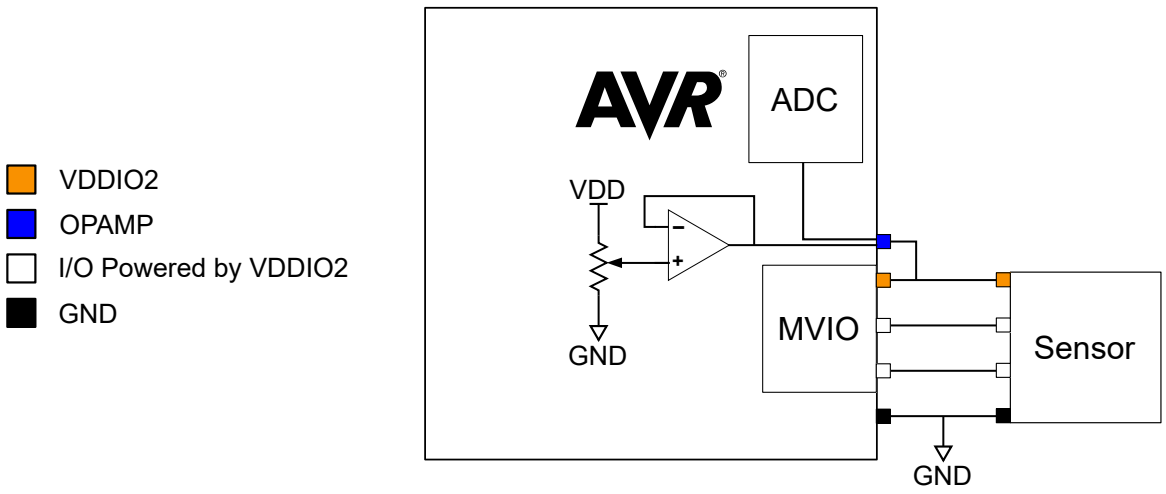
The code below sets up the ADC for use in this manner.

```
void adc_init()
{
    VREF.ADC0REF = VREF_REFSEL_4V096_gc;
    ADC0.MUXPOS = ADC_MUXPOS_AIN2_gc;
    ADC0.CTRLB= ADC_SAMPNUM_ACC16_gc;
    ADC0.CTRLA= ADC_ENABLE_bm;
}
```

To sample, we need to start an ADC conversion and read it when it is done. The following function gives this functionality.

```
uint16_t adc_sample()
{
    uint16_t res;
    ADC0.COMMAND = ADC_STCONV_bm;
    while(!(ADC0.INTFLAGS & ADC_RESRDY_bm));
    /*Right shift result by 4 due to 16 over samples*/
    res=ADC0.RES>>4;
    return res;
}
```

Figure 6-1. Monitoring the Voltage Level with the Internal ADC



The code below is based on example 5.1 Using the Internal Resistor Ladder, with added ADC functionality.

```
#define F_CPU 4000000ul
#define CLK_PER 4000000ul
#define OPAMP_MAX_SETTLE_TIME 0x7F
#include <avr/io.h>
#include <math.h>
```

```
uint16_t adc_sample();
void adc_init();
void op_amp_init();
void op_amp_setup_int_resistors(uint8_t MUXWIP_gc);

int main(void)
{
    adc_init();
    op_amp_init();
    op_amp_setup_int_resistors(OPAMP_OP0RESMUX_MUXWIP_WIP4_gc);
    while (1)
    {
        /*In this example we have output voltage of ~1.88V, check that we are within 0.1V of
that, LSB of result is 1mV*/
        uint16_t OP_out = adc_sample();
        if ((1780<OP_out) & (OP_out<1980))
        {
            /*Your code goes here*/
        }
    }
}


uint16_t adc_sample()
{
    uint16_t res;
    ADC0.COMMAND = ADC_STCONV_bm;
    while(!(ADC0.INTFLAGS & ADC_RESRDY_bm));
    /*Right shift result by 4 due to 16 over samples*/
    res=ADC0.RES>>4;
    return res;
}

void adc_init()
{
    VREF.ADC0REF = VREF_REFSEL_4V096_gc;
    ADC0.MUXPOS = ADC_MUXPOS_AIN2_gc;
    ADC0.CTRLB= ADC_SAMPNUM_ACC16_gc;
    ADC0.CTRLA= ADC_ENABLE_bm;
}

void op_amp_init()
{
    /*Disable input on op amp output pin*/
    PORTD.PIN2CTRL = PORT_ISC_INPUT_DISABLE_gc;

    /*Set up op amp*/
    OPAMP.CTRLA = OPAMP_ENABLE_bm;
    OPAMP.TIMEBASE = (uint8_t) ceil(CLK_PER*0.000001)-1; /*Number of peripheral clock cycles
that amounts to 1us*/
    OPAMP.OP0CTRLA = OPAMP_RUNSTBY_bm | OPAMP_ALWAYSON_bm | OPAMP_OP0CTRLA_OUTMODE_NORMAL_gc;
    OPAMP.OP0SETTLE = OPAMP_MAX_SETTLE_TIME; //As the settle time is unknown, the maximums
should be set
}

void op_amp_setup_int_resistors(OPAMP_OP0RESMUX_MUXWIP_t MUXWIP_gc)
{
    OPAMP.OP0INMUX = OPAMP_OP0INMUX_MUXNEG_OUT_gc | OPAMP_OP0INMUX_MUXPOS_WIP_gc;
    /* MUXWIP is the bit field that decides the output voltage of the op amp
       With VDD at 5V you will get:
       OPAMP_OP0RESMUX_MUXWIP_WIP4_gc -> ~1.88V
       OPAMP_OP0RESMUX_MUXWIP_WIP3_gc -> ~2.5V
       OPAMP_OP0RESMUX_MUXWIP_WIP2_gc -> ~3.75V */
    OPAMP.OP0RESMUX = OPAMP_OP0RESMUX_MUXTOP_VDD_gc|OPAMP_OP0RESMUX_MUXBOT_GND_gc|MUXWIP_gc;
}
```

The code for this example is available in the **monitoring-OPAMP-output-with-ADC** folder in these github repositories:

7. References

1. AVR128DB48 product page: www.microchip.com/wwwproducts/en/AVR128DB48.
2. AVR128DB48: www.microchip.com/DS40002247.
3. Curiosity Nano AVR128DB48 product page: www.microchip.com/DevelopmentTools/ProductDetails/PartNO/EV35L43A.
4. AVR128DB48 Curiosity Nano User Guide: www.microchip.com/DS50003037.
5. Curiosity Nano AVR128DB48 Schematics ww1.microchip.com/downloads/en/DeviceDoc/AVR128DB48_Curiosity_Nano_Schematics.pdf.

8. Revision History

Doc. Rev.	Date	Comments
A	09/2020	Initial document release.

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods being used in attempts to breach the code protection features of the Microchip devices. We believe that these methods require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Attempts to breach these code protection features, most likely, cannot be accomplished without violating Microchip's intellectual property rights.
- Microchip is willing to work with any customer who is concerned about the integrity of its code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "unbreakable." Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Trademarks

ISBN: 978-1-5224-6664-2

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: www.microchip.com/support Web Address: www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4485-5910 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-72400 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820