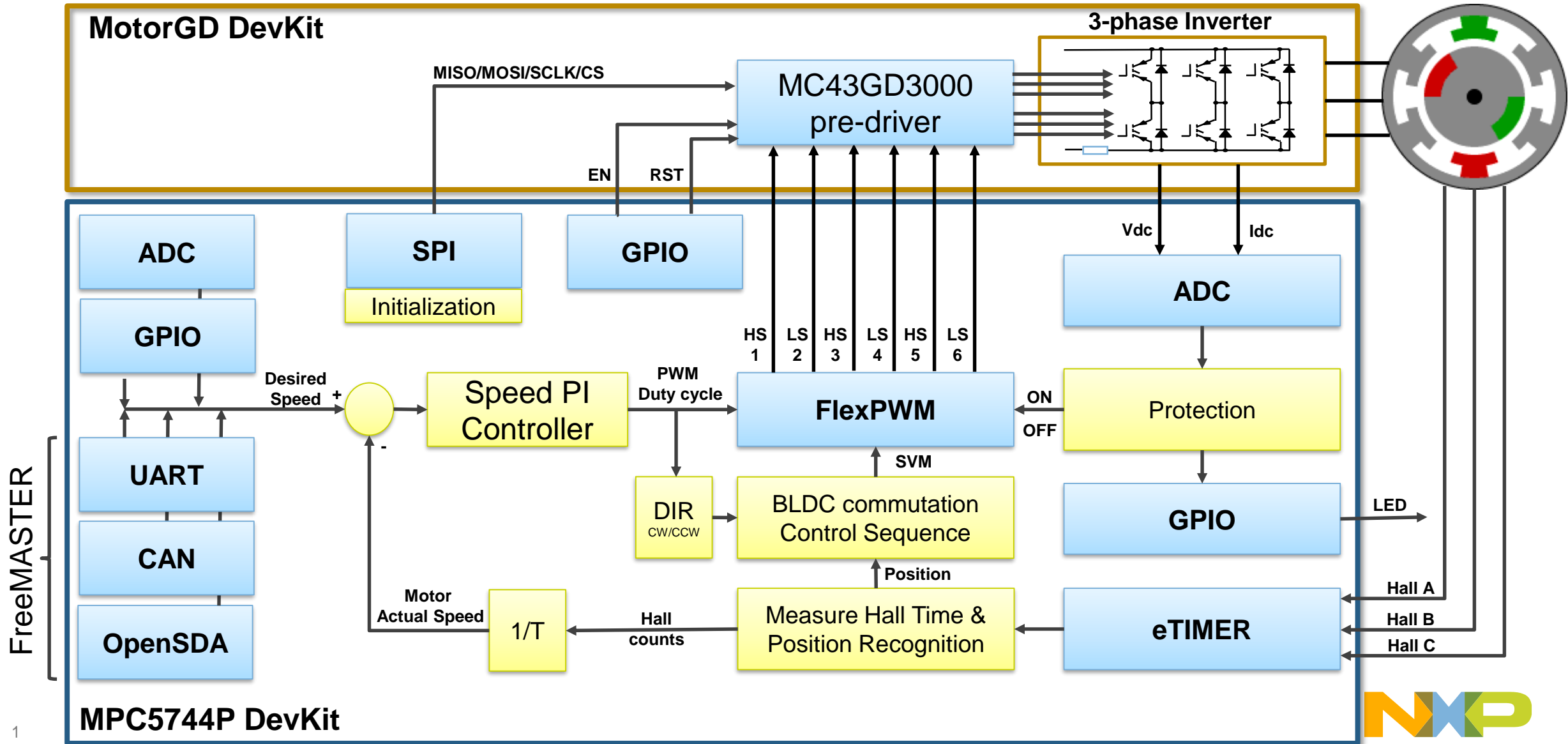


# MODEL BASED DESIGN TOOLBOX ENABLING FAST PROTOTYPING AND DESIGN

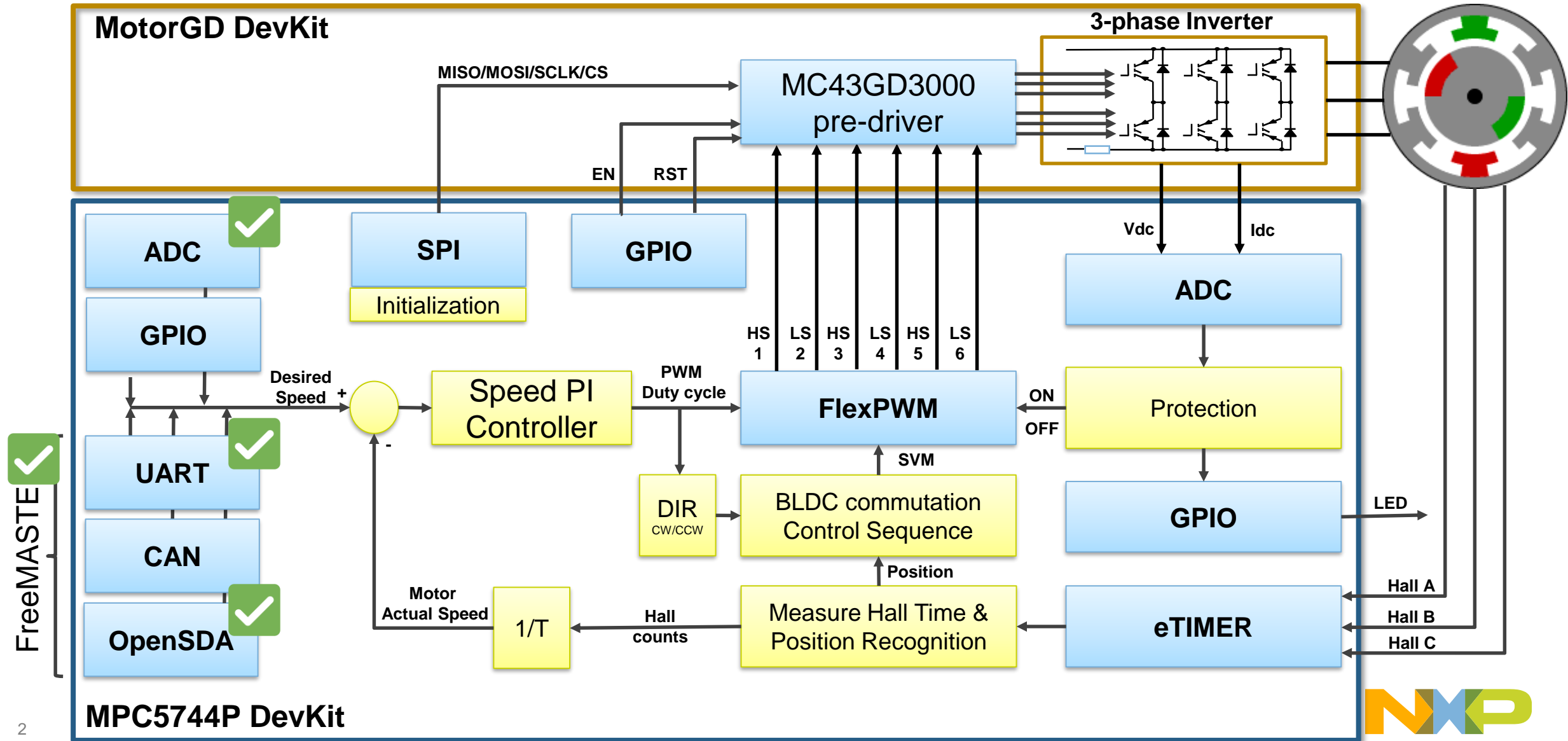
- MCU RBF programming
- ADC how it works
- ADC configuration and testing
- Model validation via FreeMASTER



# Application mapping: SW vs. HW



# Application mapping: SW vs. HW



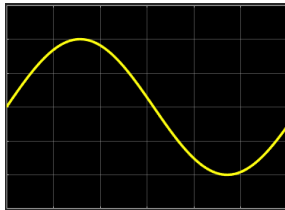
# ADC – How it works

- In general there are 2 main stages:
  1. **S/H – Sampling and Holding:** is an analog device that samples (captures, grabs) the voltage of a continuously varying analog signal and holds (locks, freezes) its value at a constant level for a specified minimum period of time
  2. **Q/E – Quantizing and Encoding:** the nearest value, in correspondence with the amplitude of sampling and holding signal is transformed via another analog device into binary output

# ADC – How it works

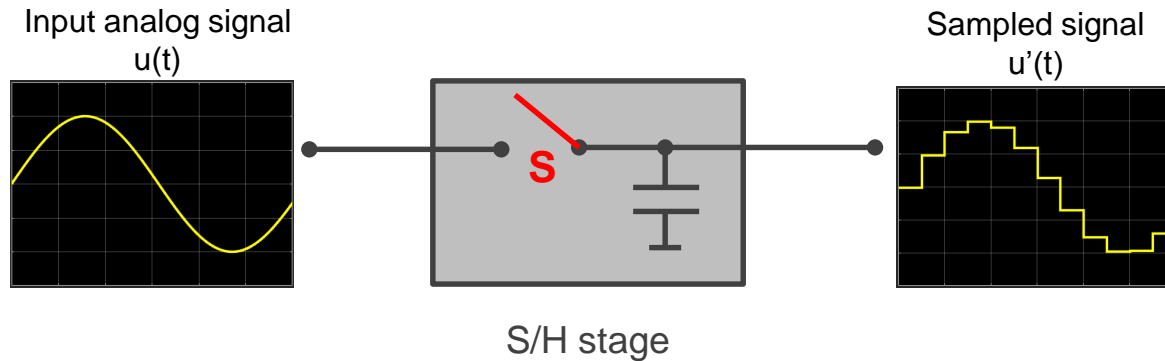
- In general there are 2 main stages:
  1. **S/H – Sampling and Holding:** is an analog device that samples (captures, grabs) the voltage of a continuously varying analog signal and holds (locks, freezes) its value at a constant level for a specified minimum period of time
  2. **Q/E – Quantizing and Encoding:** the nearest value, in correspondence with the amplitude of sampling and holding signal is transformed via another analog device into binary output

Input analog signal  
 $u(t)$



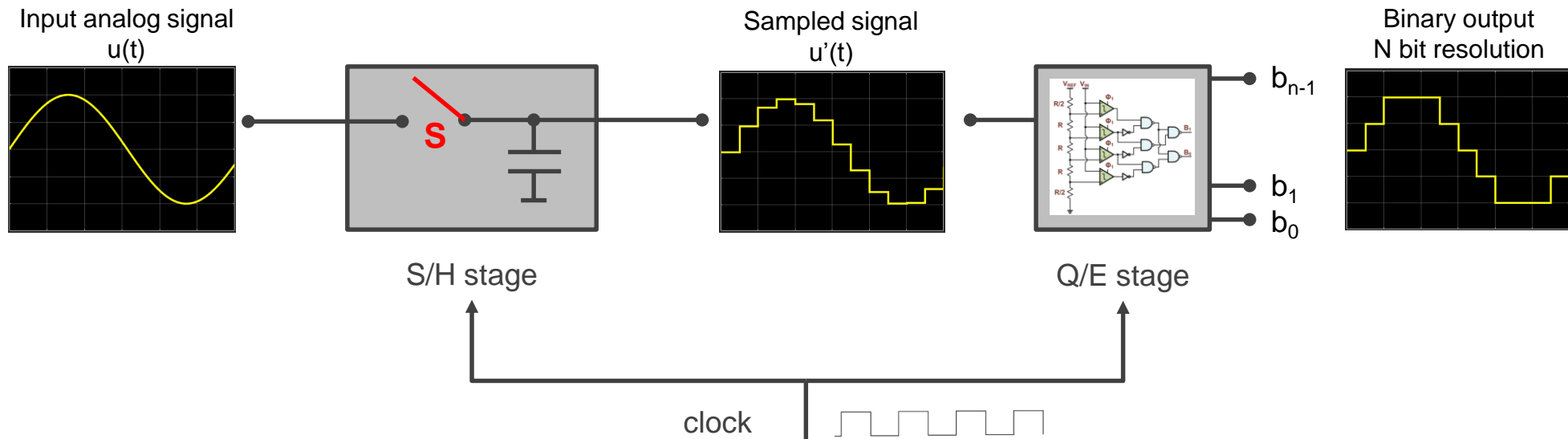
# ADC – How it works

- In general there are 2 main stages:
  1. **S/H – Sampling and Holding:** is an analog device that samples (captures, grabs) the voltage of a continuously varying analog signal and holds (locks, freezes) its value at a constant level for a specified minimum period of time
  2. **Q/E – Quantizing and Encoding:** the nearest value, in correspondence with the amplitude of sampling and holding signal is transformed via another analog device into binary output



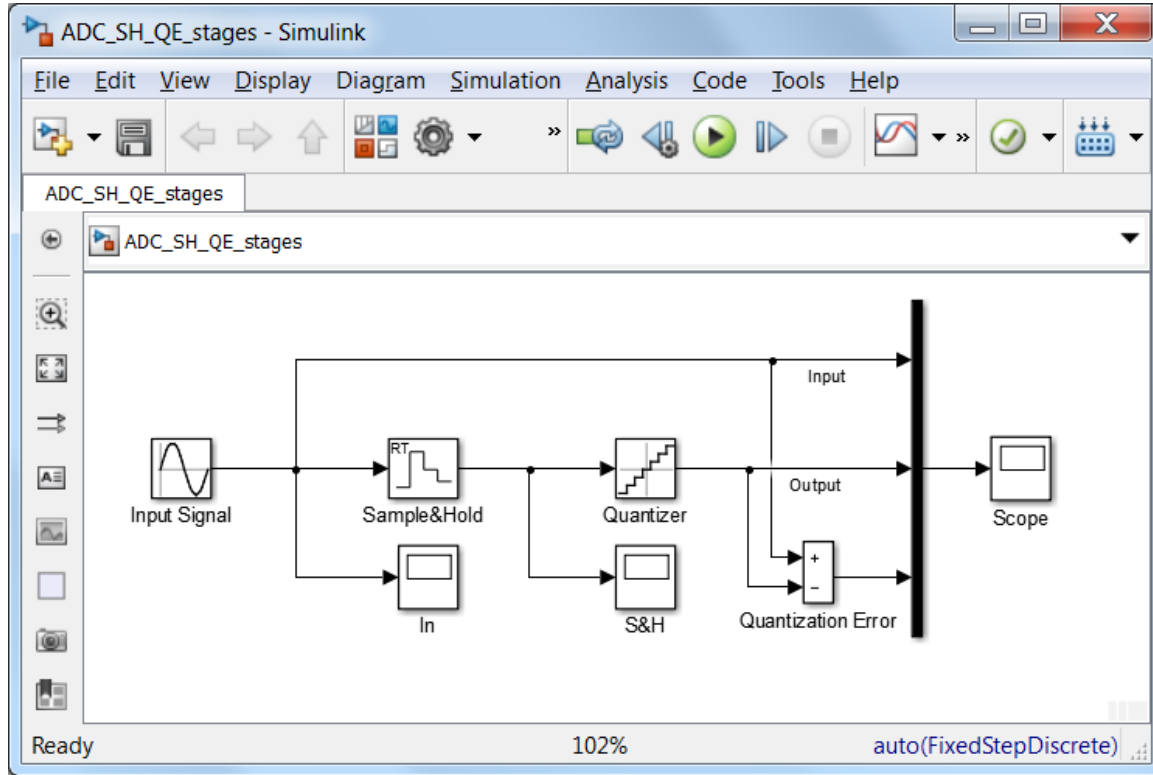
# ADC – How it works

- In general there are 2 main stages:
  1. **S/H – Sampling and Holding:** is an analog device that samples (captures, grabs) the voltage of a continuously varying analog signal and holds (locks, freezes) its value at a constant level for a specified minimum period of time
  2. **Q/E – Quantizing and Encoding:** the nearest value, in correspondence with the amplitude of sampling and holding signal is transformed via another analog device into binary output



# ADC – Modelling

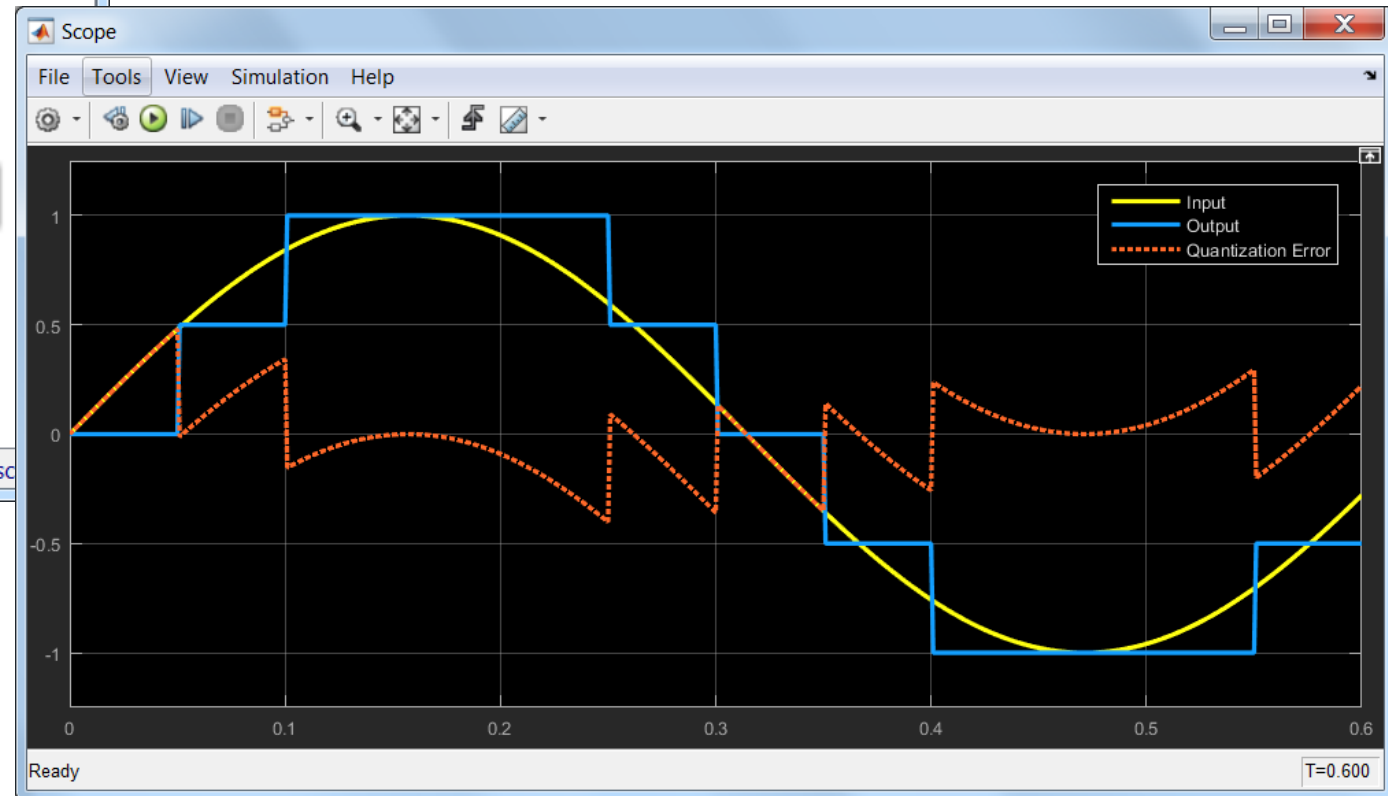
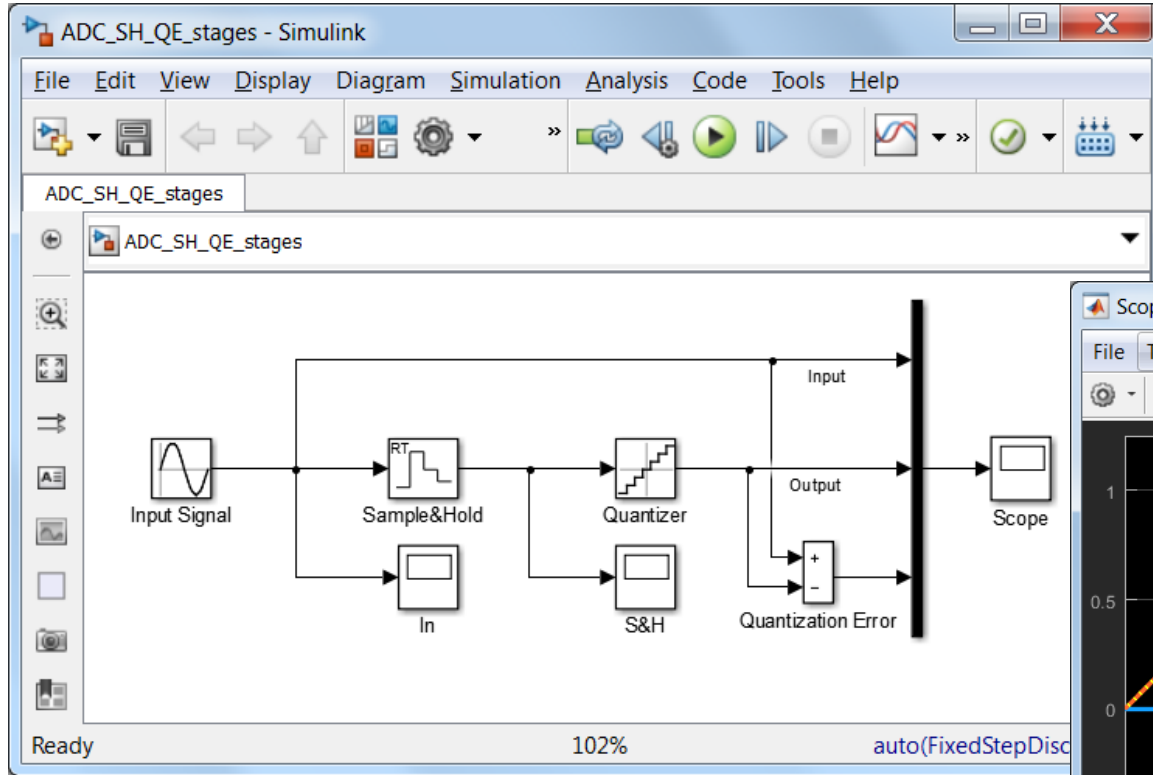
Can be easily modelled in Simulink via ZOH and Quantizer standard blocks





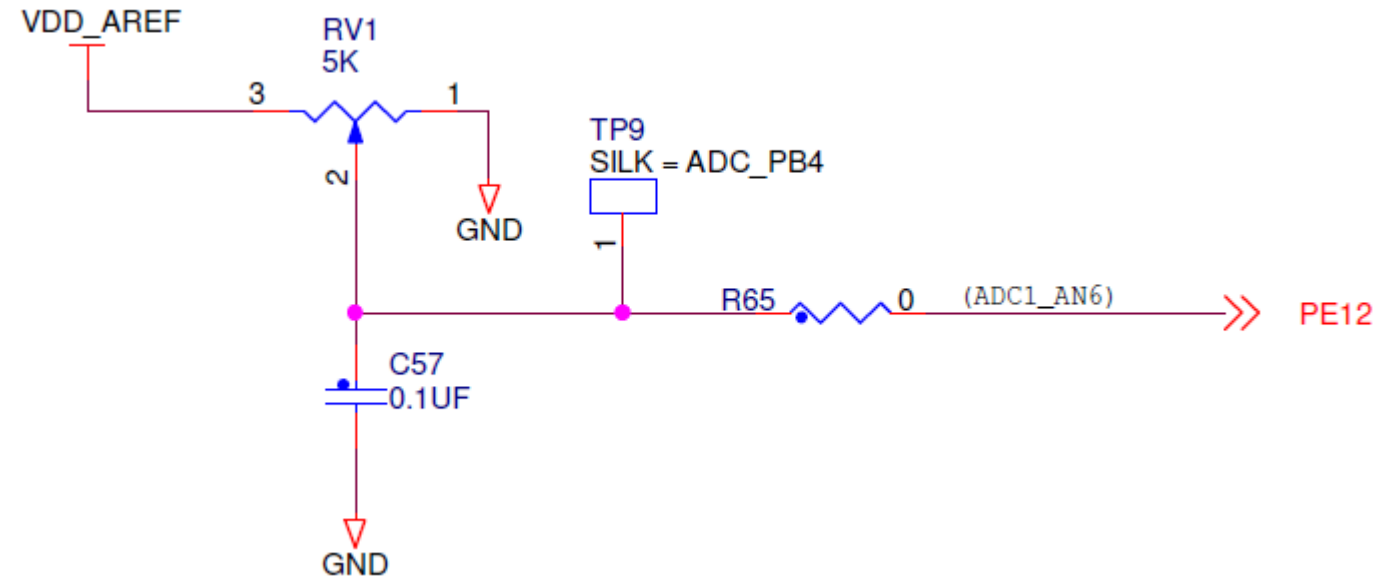
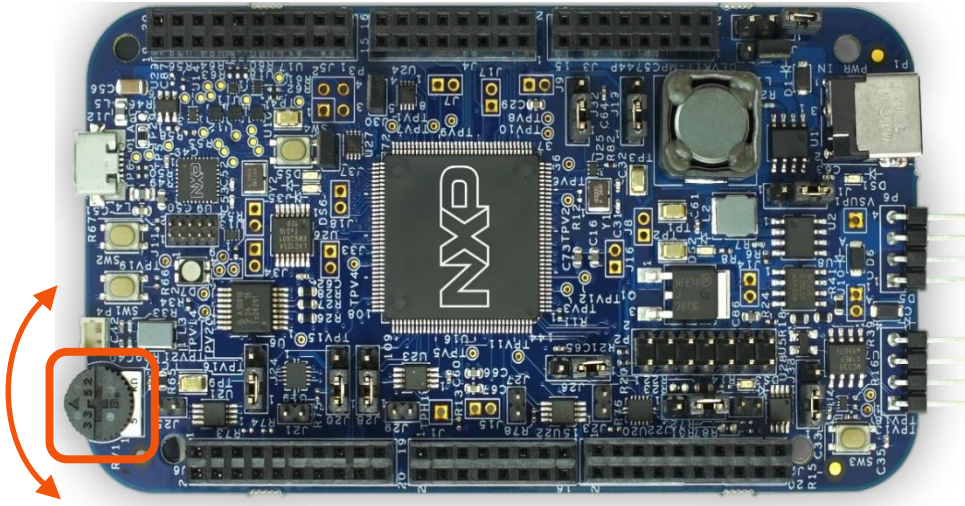
# ADC – Modelling

Can be easily modelled in Simulink via ZOH and Quantizer standard blocks



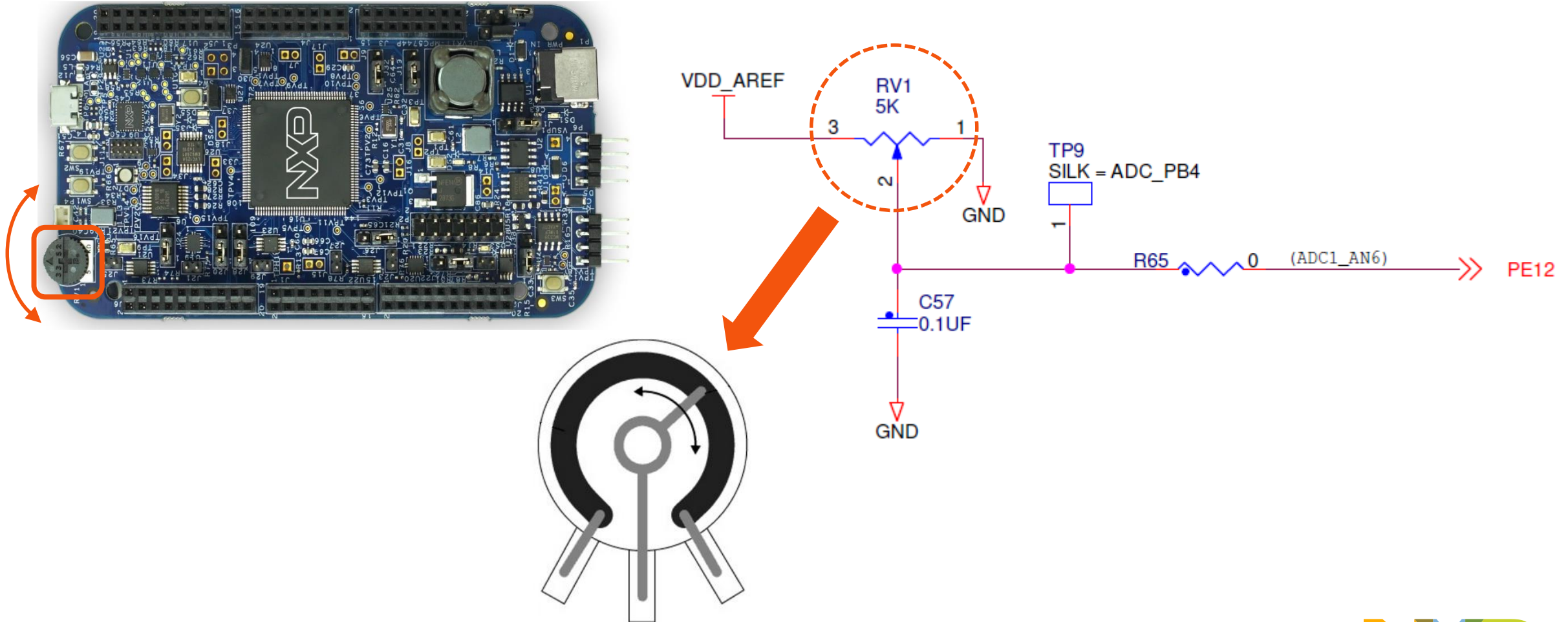
# ADC – On board Potentiometer

MPC5744P DevKit has an onboard Potentiometer what we shall use for analog speed reference



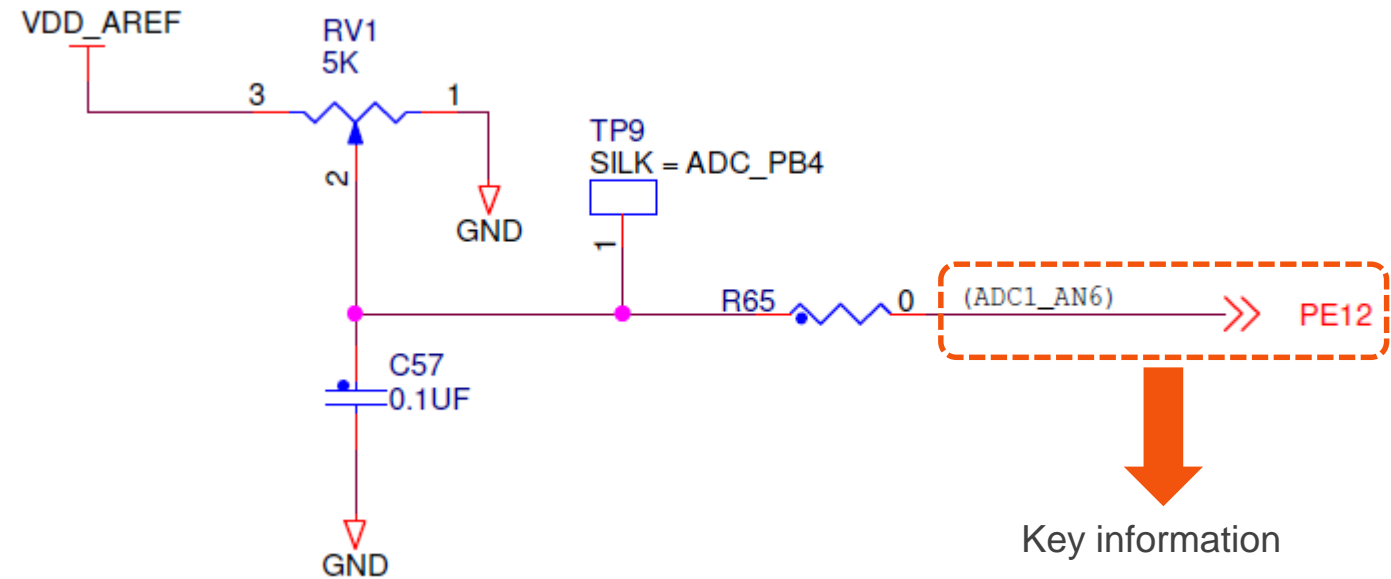
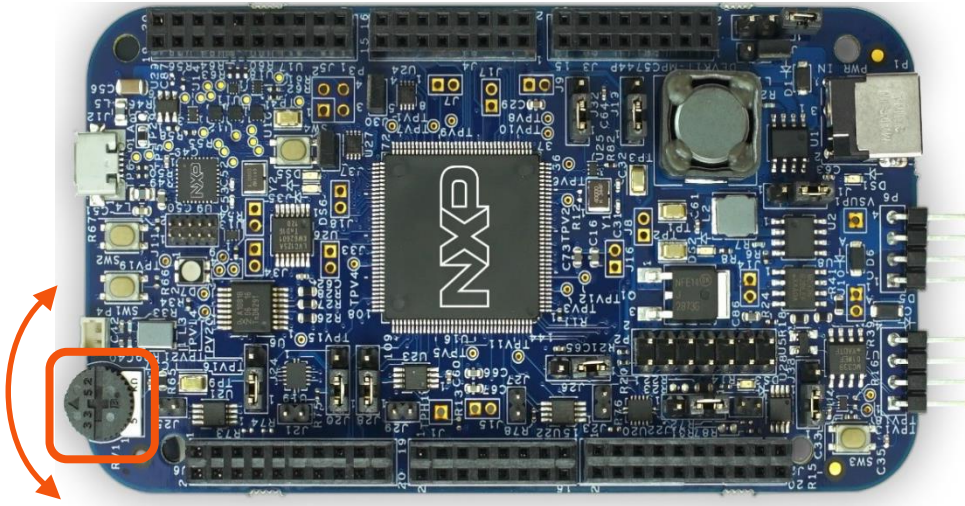
# ADC – On board Potentiometer

MPC5744P DevKit has an onboard Potentiometer what we shall use for analog speed reference



# ADC – On board Potentiometer

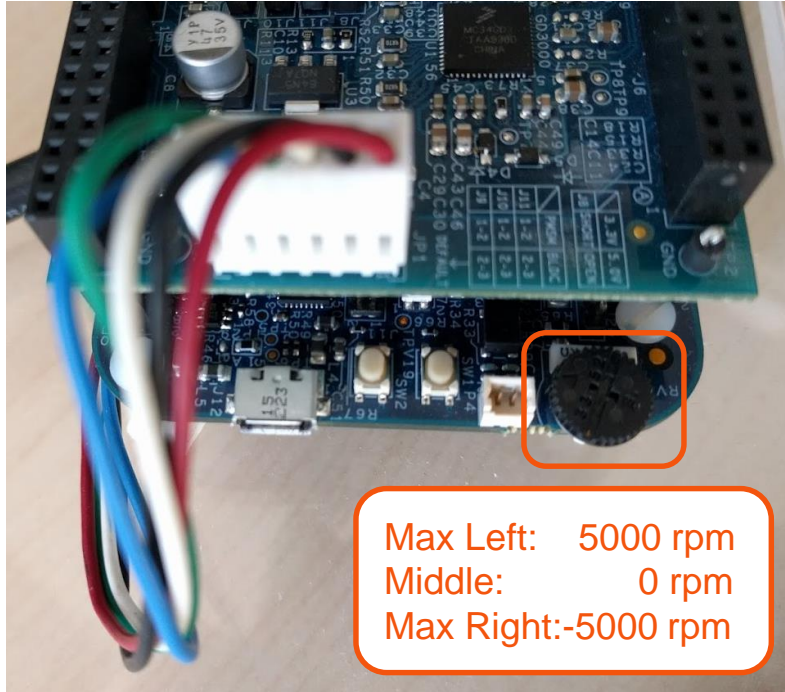
MPC5744P DevKit has an onboard Potentiometer what we shall use for analog speed reference



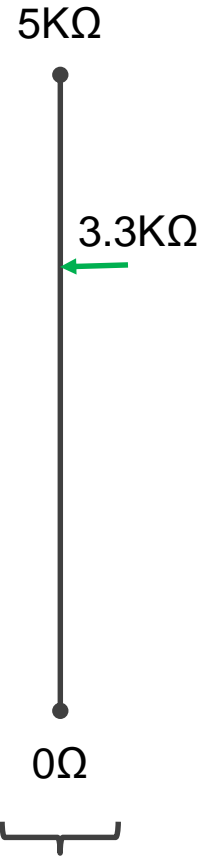
Key information

ADC module 1  
ADC channel 6  
CPU Pin **67** @144LQFP

# Convert Potentiometer value into Desired Speed Reference

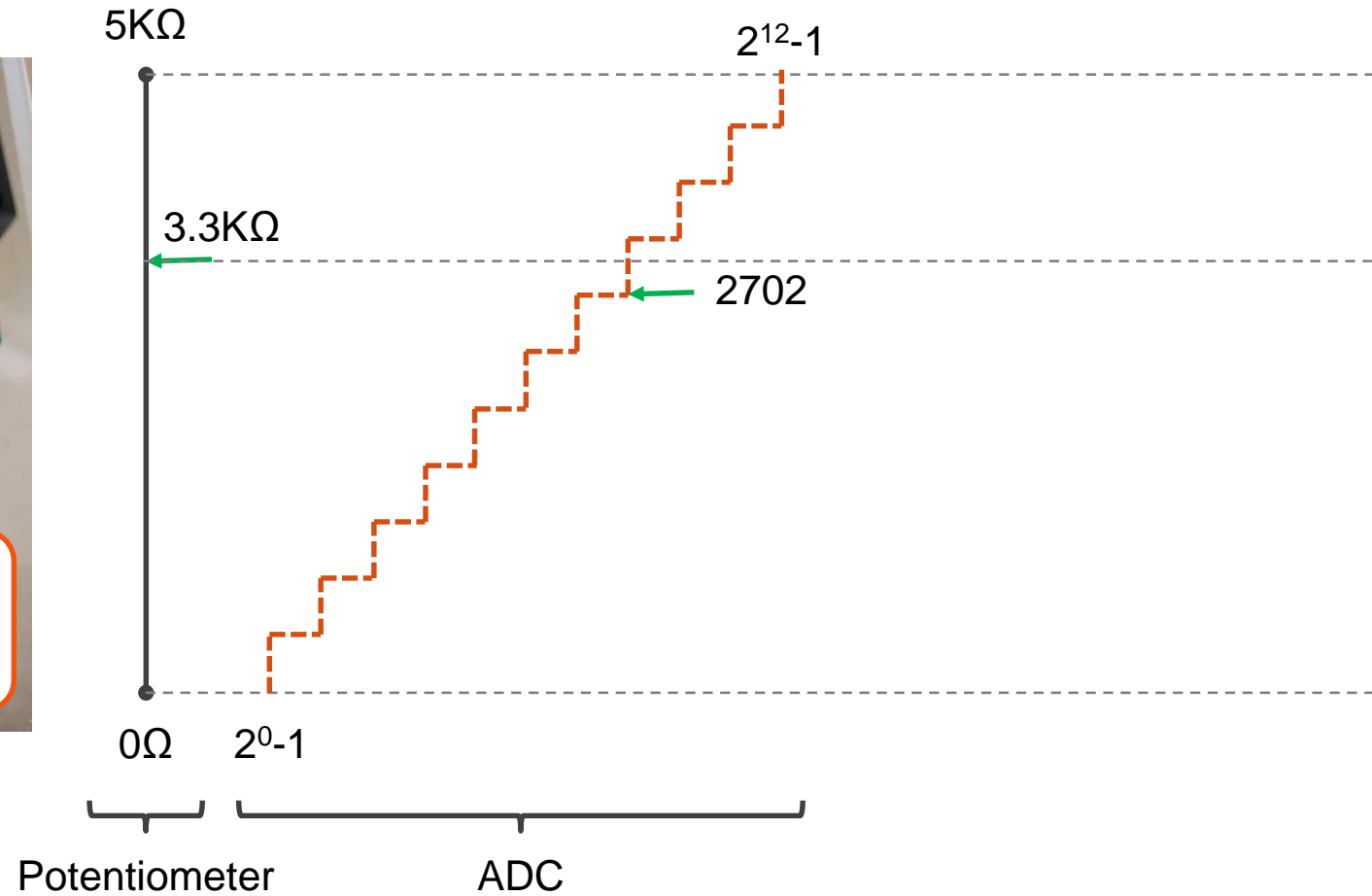
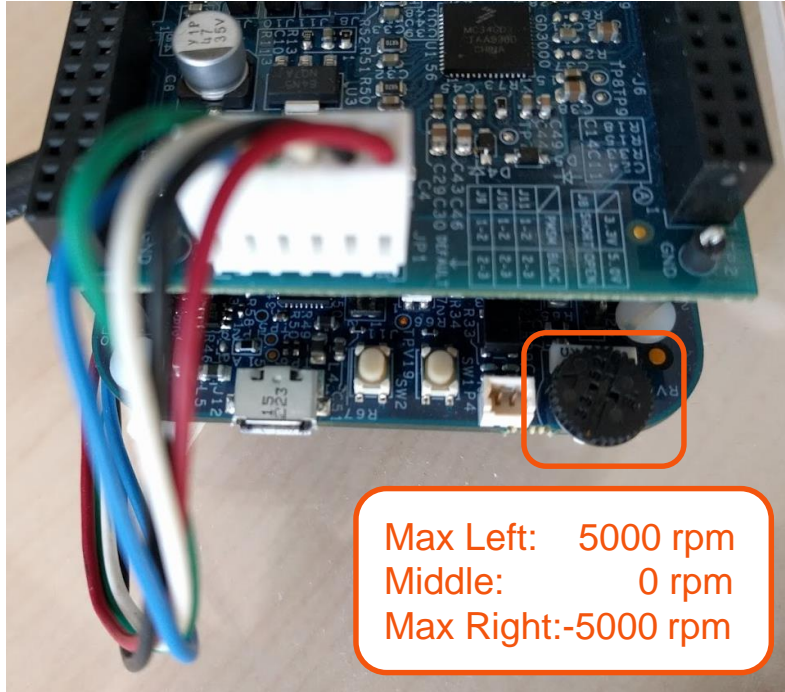


Max Left: 5000 rpm  
Middle: 0 rpm  
Max Right: -5000 rpm

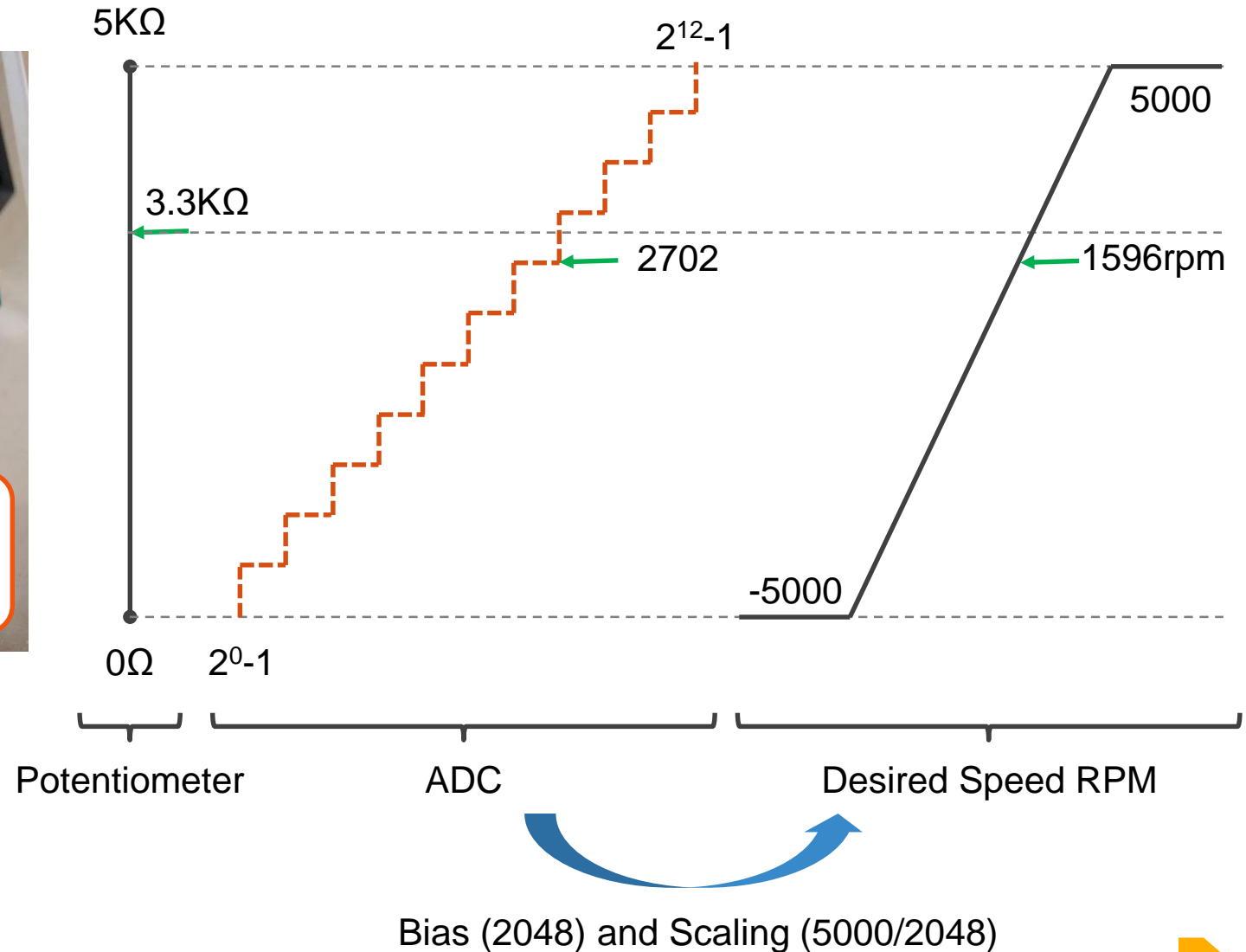
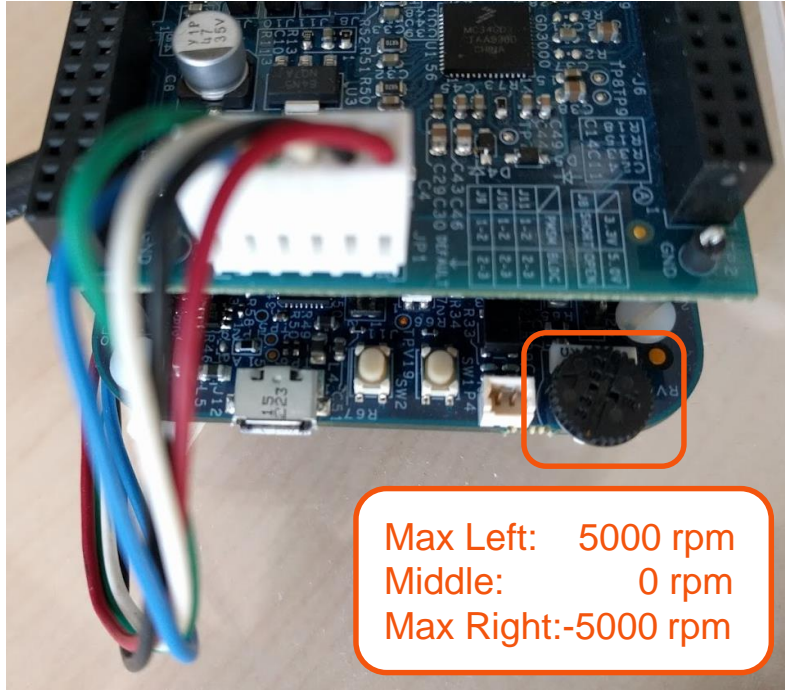


Potentiometer

# Convert Potentiometer value into Desired Speed Reference



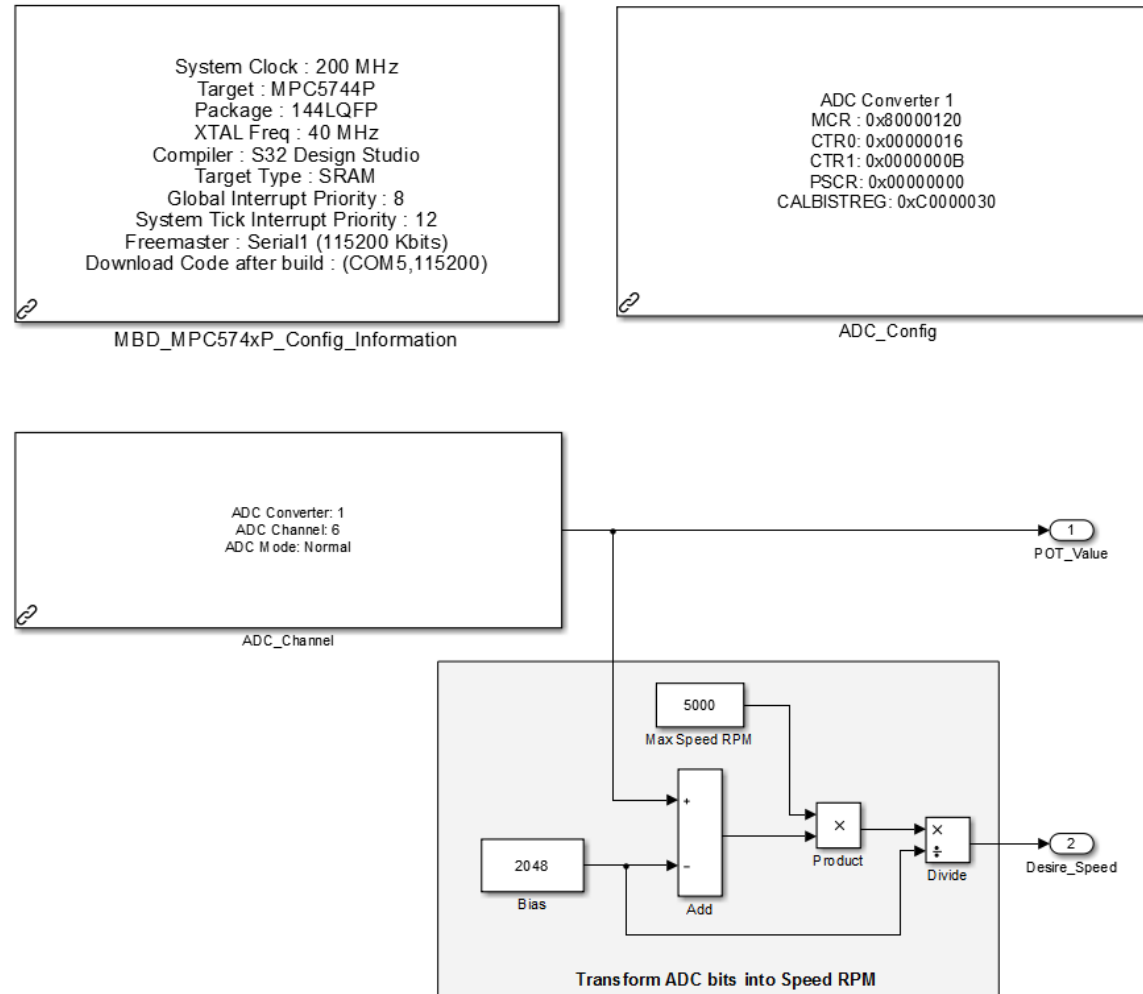
# Convert Potentiometer value into Desired Speed Reference



# ADC model

## Options:

- Read ADC continuously.
- Convert ADC value into Speed RPM units.
- Use FreeMASTER validate ADC conversions in real time.
- All computations made on fixed point.





# MODEL BASED DESIGN TOOLBOX ENABLING FAST PROTOTYPING AND DESIGN

...you are now ready to design the next big thing. Good luck!

Need help? Visit us at: <https://community.nxp.com/community/mbdt> to share ideas

