

90_BasicSignals

March 22, 2020

1 Generate Basic Signals

If your MDF files are locked away, proprietary, or otherwise unavailable. Generate simple signals, save them to a MDF file and re-open them for analysis.

Generate basic controls signals:

```
[1]: import mdngen
     for signal_generator in mdngen.signal_generators:
         print(signal_generator.__name__)
```

```
sine
cosine
square
sawtooth
triangle
pwm
step
```

```
[2]: import IPython.display as display
     import matplotlib.pyplot as plt
     import numpy as np
     import pandas as pd
     import seaborn as sns
```

```
[3]: tf = 10 # [s]
     f = 100 # [Hz]
     # Multiple time vectors
     t = np.arange(0, tf, 1/f, dtype=np.float32)
```

```
[4]: from asammdf import Signal
```

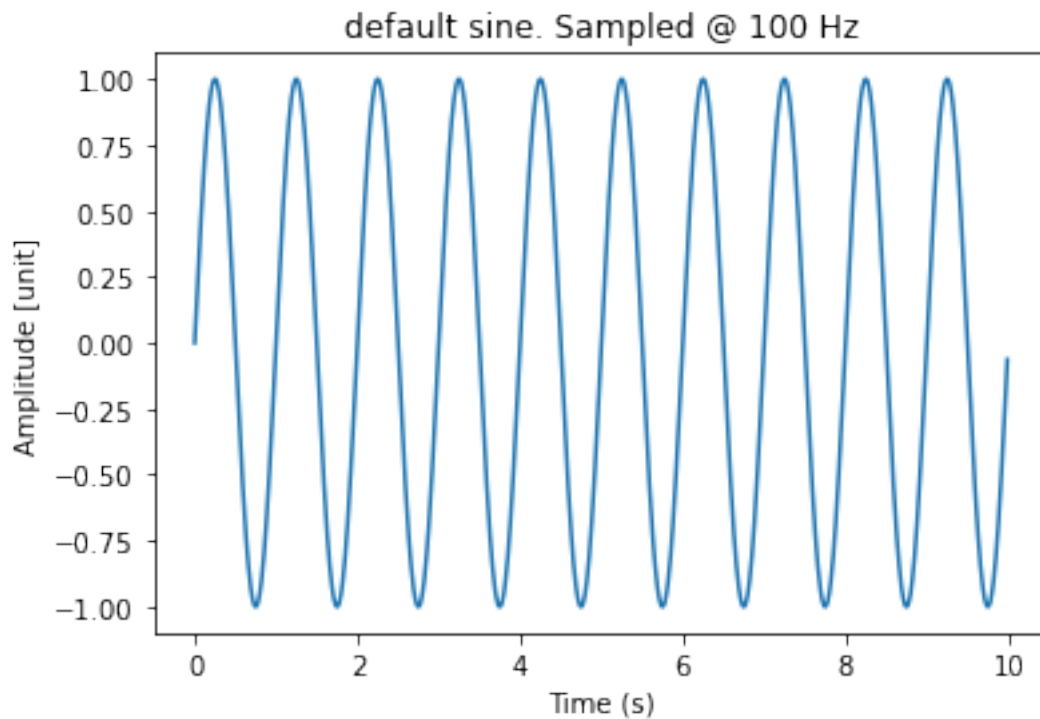
Plot the signal with matplotlib.

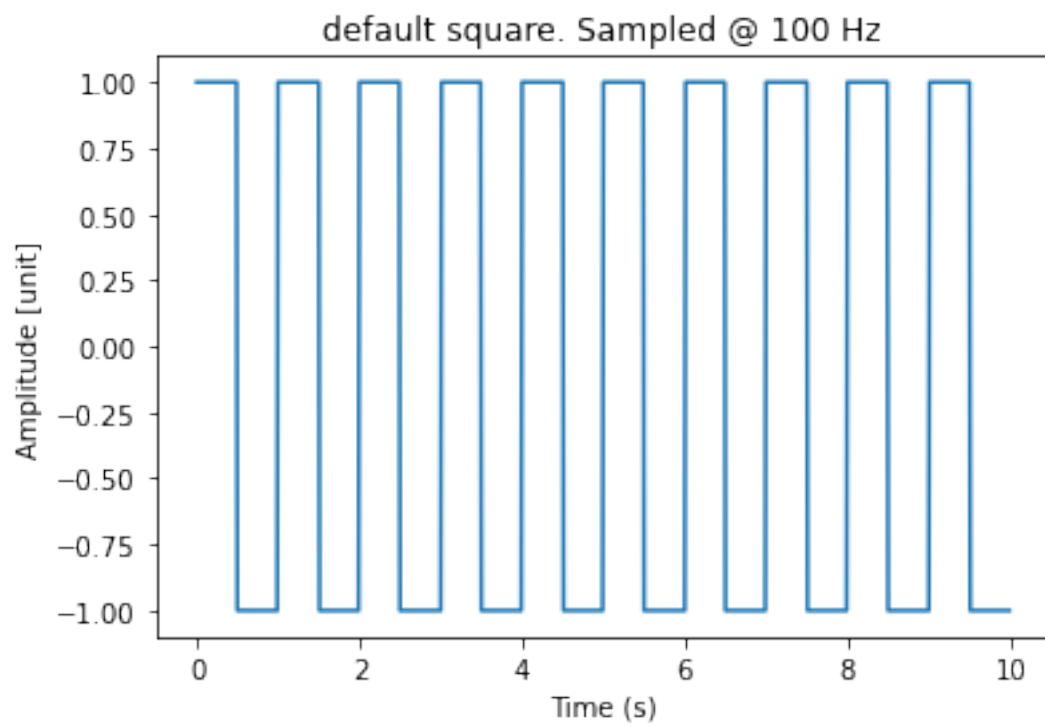
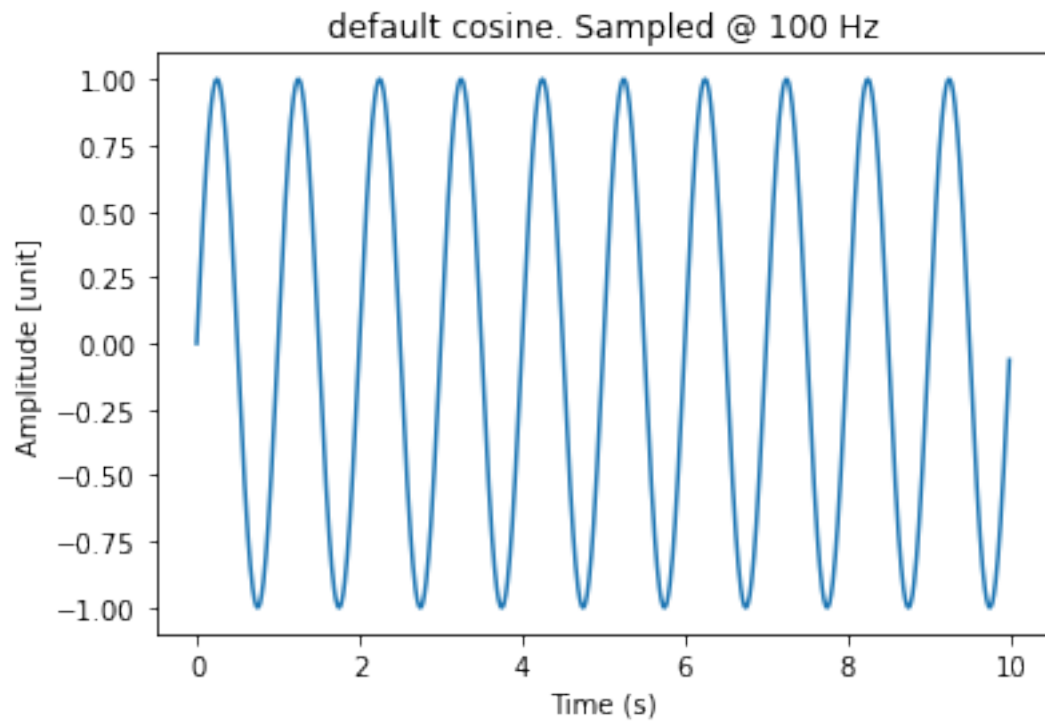
```
[5]: sigs = list()
     for signal_generator in mdngen.signal_generators:
         plt.figure()
         y = signal_generator(t)
```

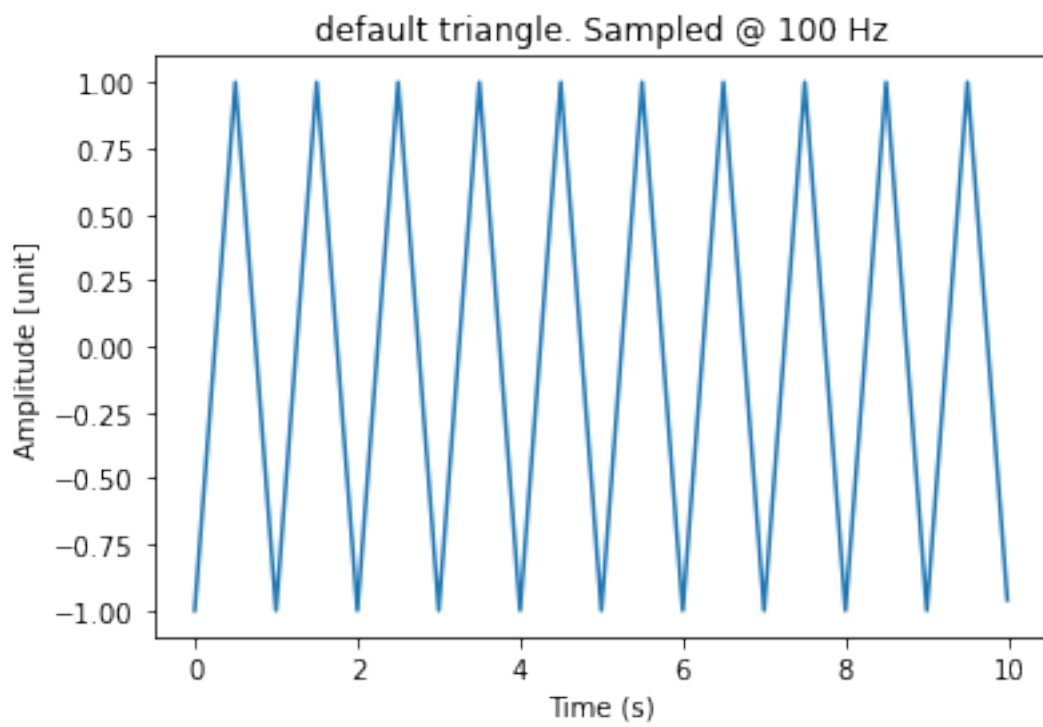
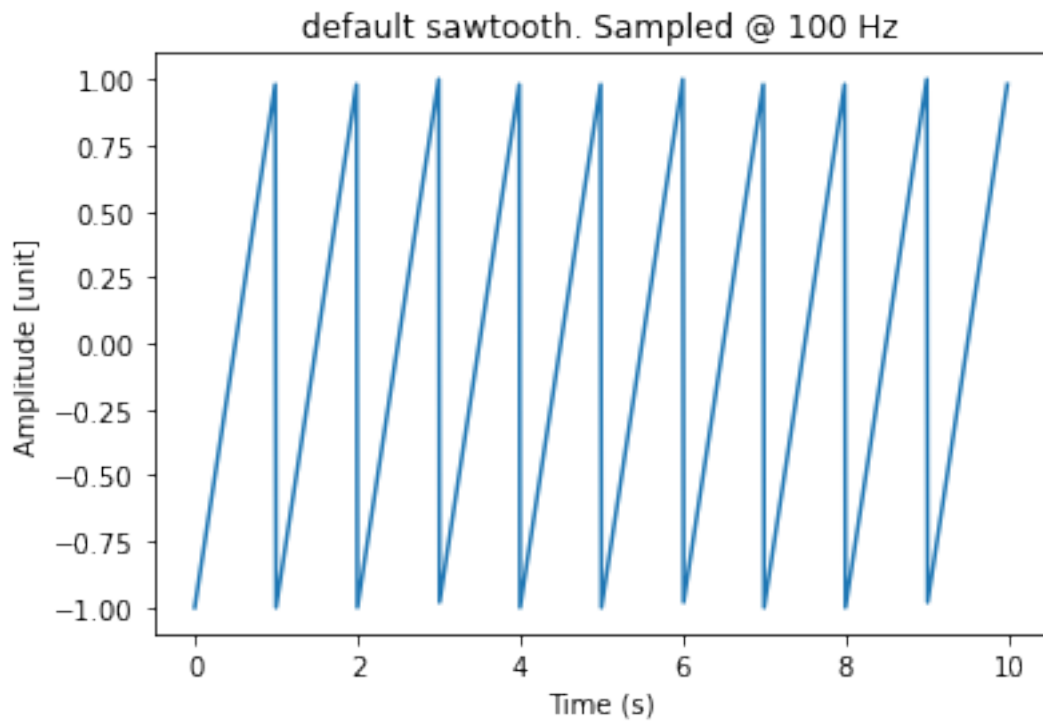
```

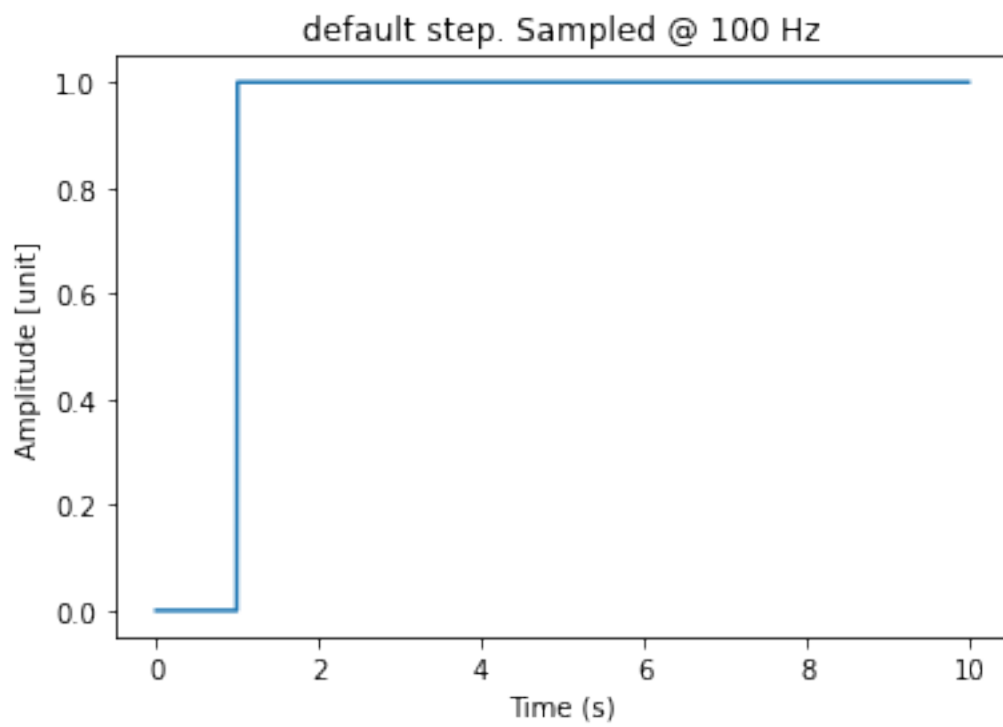
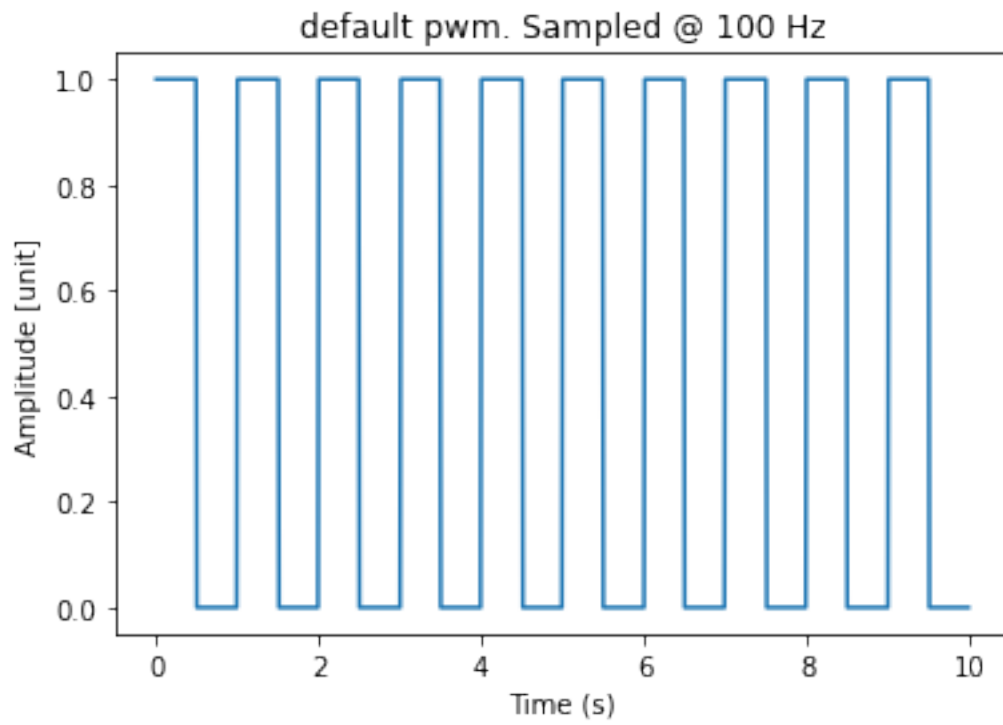
sig = Signal(
    y,
    t,
    name=signal_generator.__name__,
    unit=None,
    conversion=None,
    comment=f"default {signal_generator.__name__}",
)
sigs.append(sig)
# Method 1.
plt.plot(t, y)
plt.xlabel("Time (s)")
plt.ylabel("Amplitude [unit]")
plt.title(f"default {signal_generator.__name__}. Sampled @ 100 Hz")
plt.savefig(f"default_{signal_generator.__name__}.png", transparent=False,
    ↳bbox_inches='tight')

```





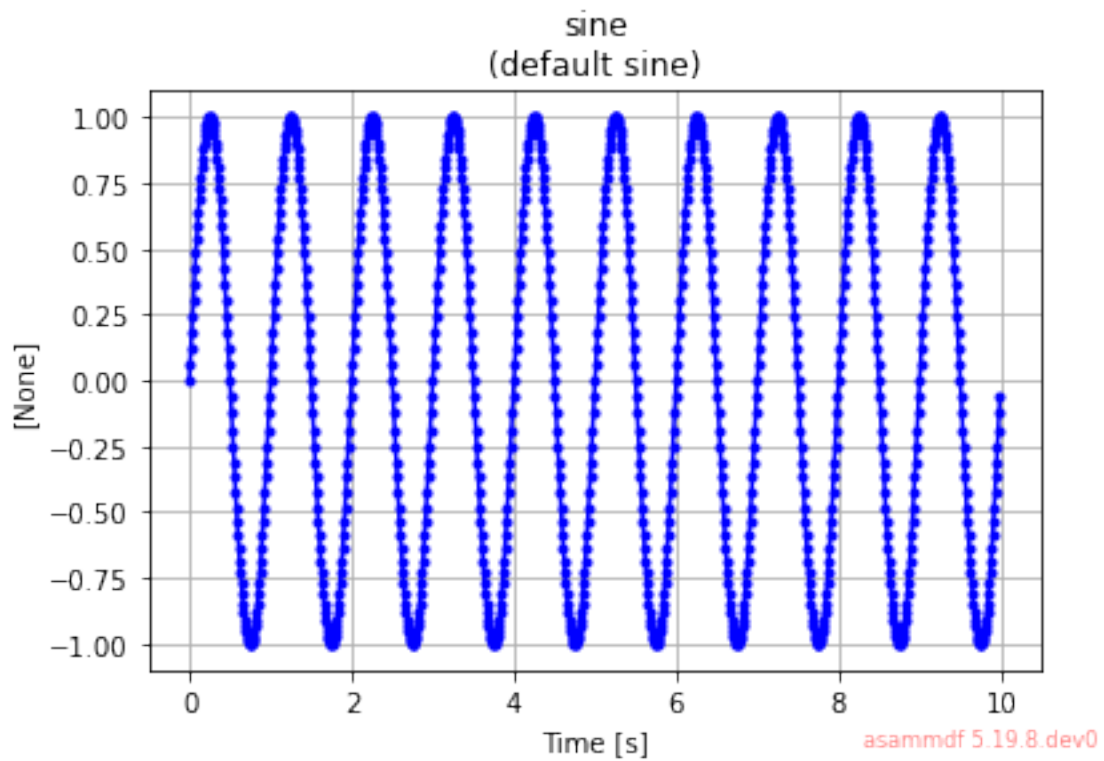




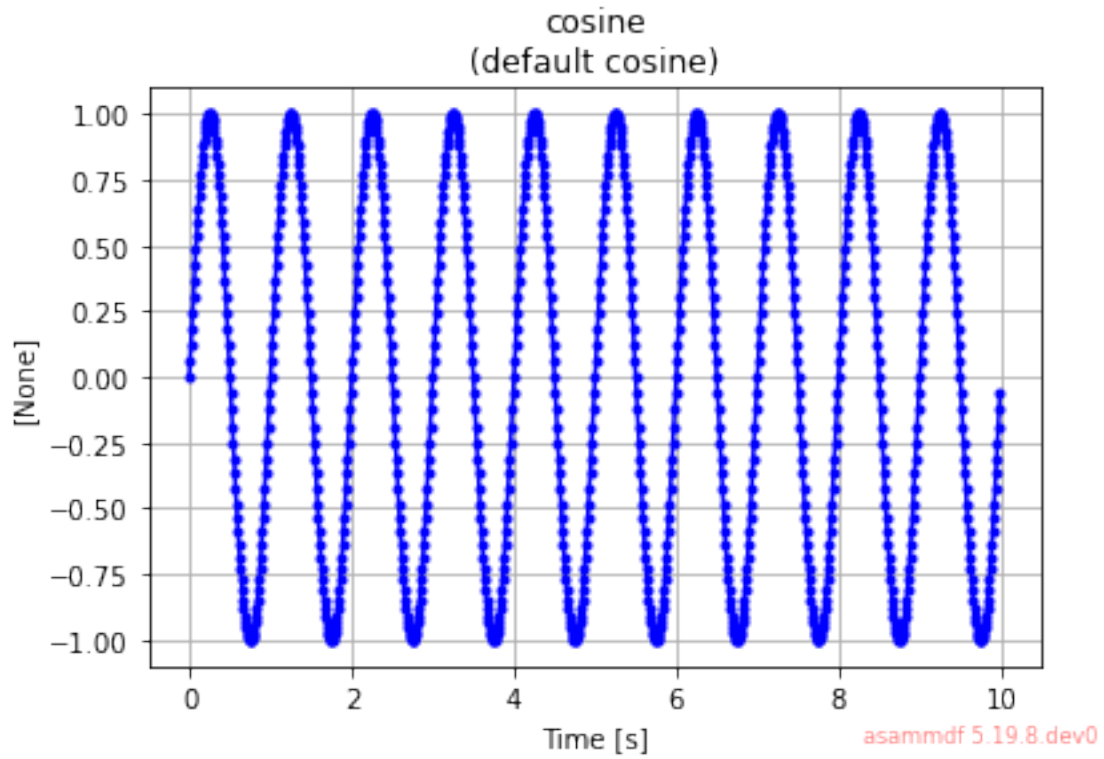
1.1 Plot the signals with asammdf

```
[6]: for sig in sigs:  
      sig.plot()
```

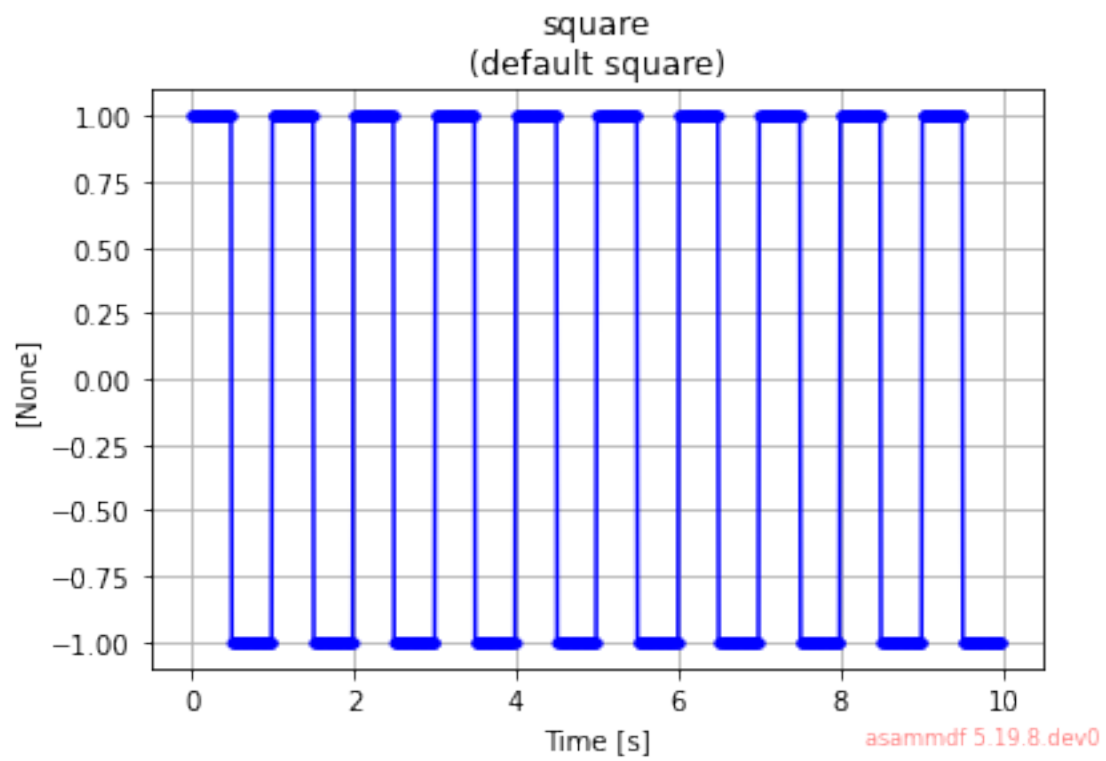
WARNING:root:Signal plotting requires pyqtgraph or matplotlib



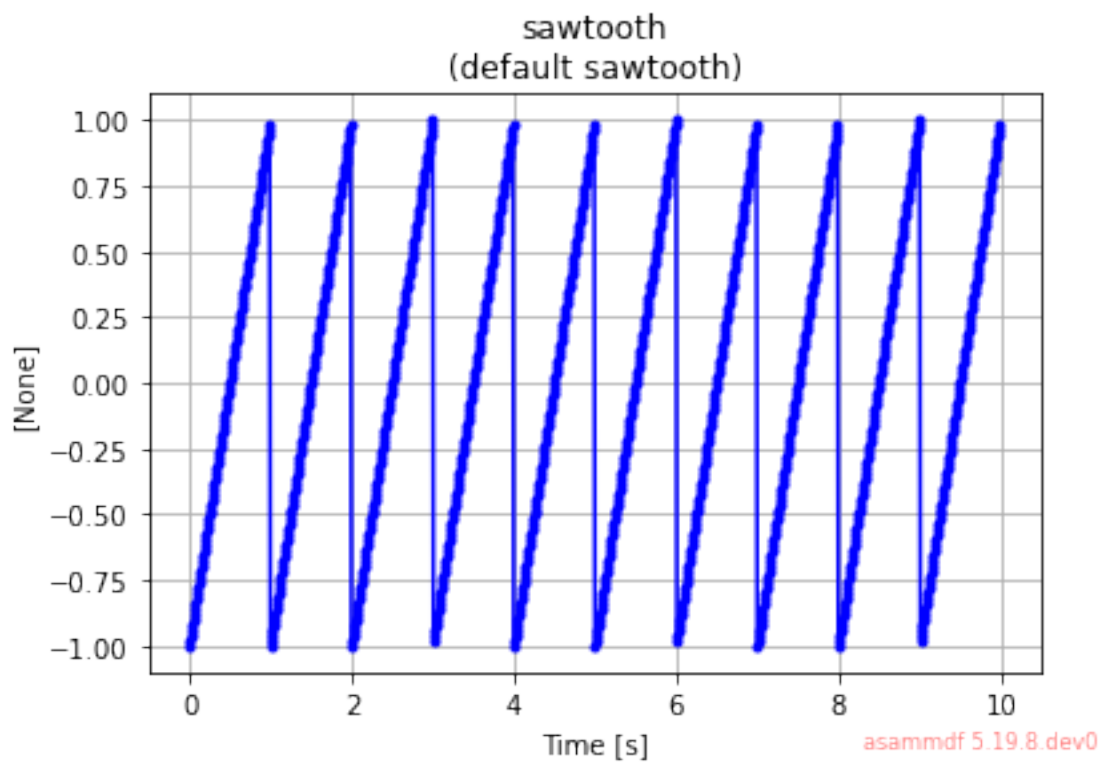
WARNING:root:Signal plotting requires pyqtgraph or matplotlib



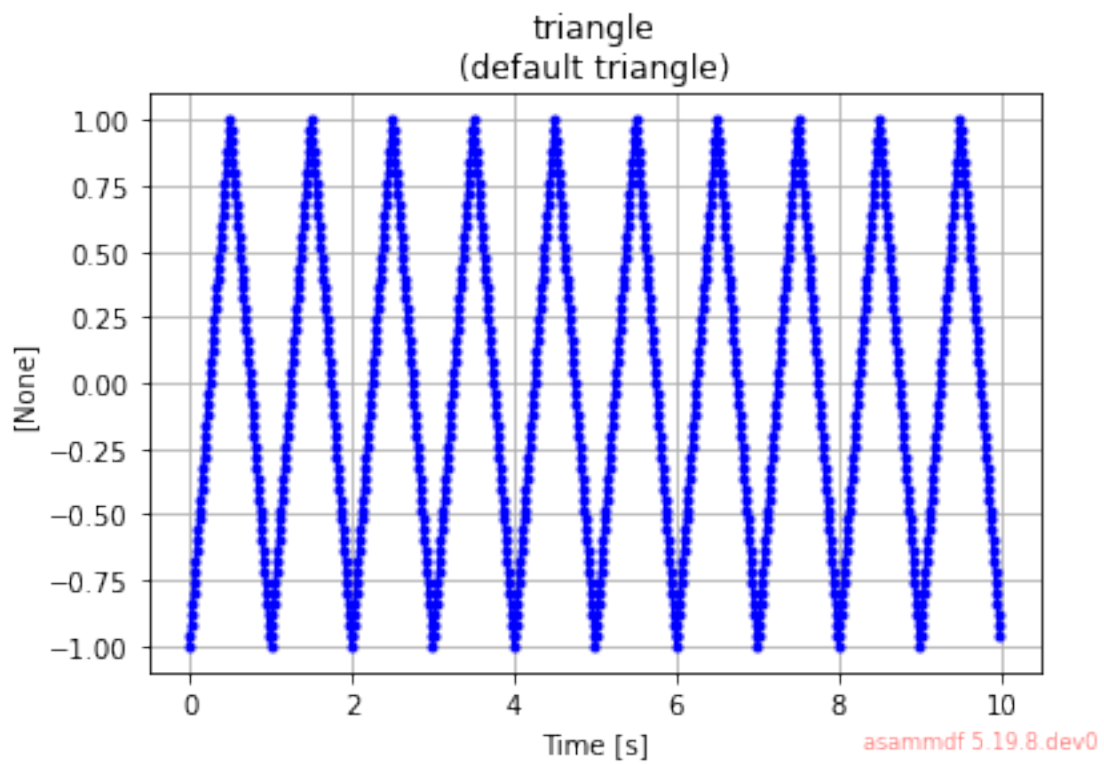
WARNING:root:Signal plotting requires pyqtgraph or matplotlib



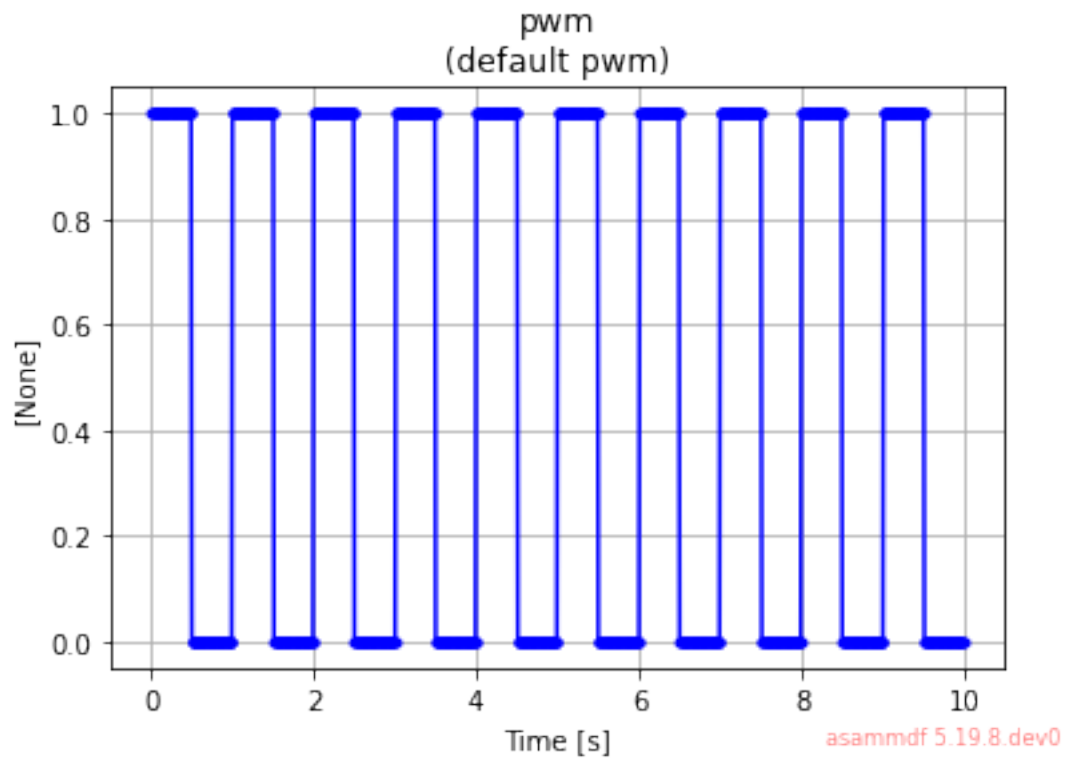
WARNING:root:Signal plotting requires pyqtgraph or matplotlib



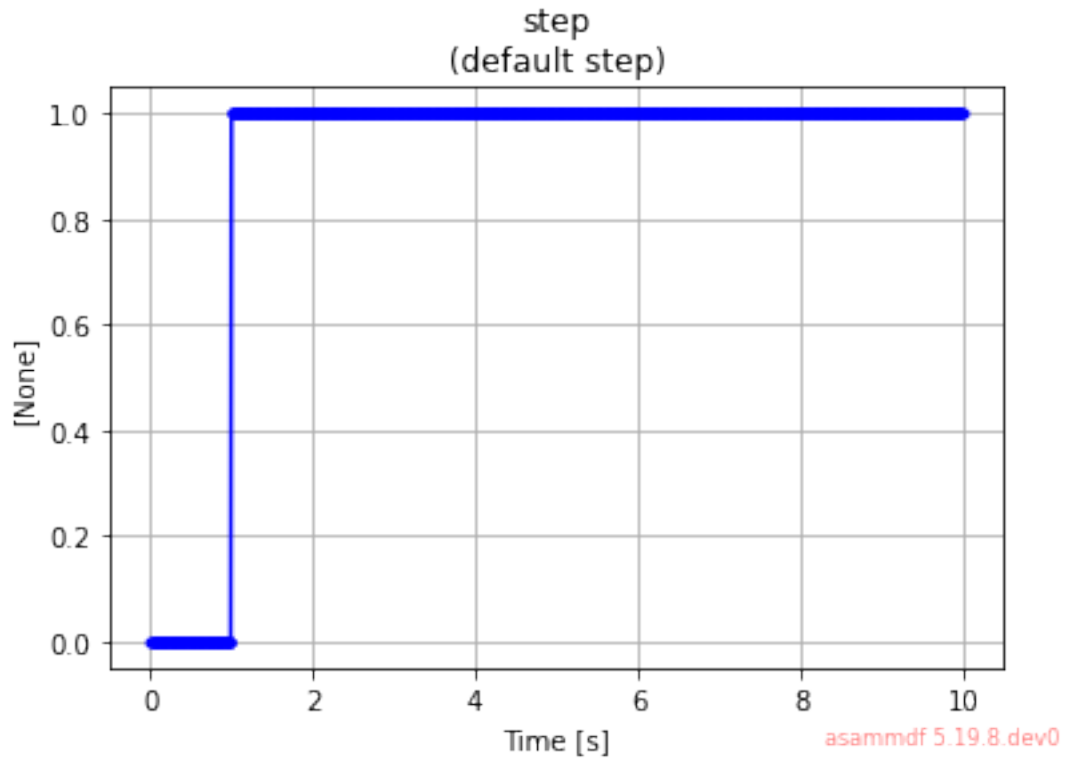
WARNING:root:Signal plotting requires pyqtgraph or matplotlib



WARNING:root:Signal plotting requires pyqtgraph or matplotlib



WARNING:root:Signal plotting requires pyqtgraph or matplotlib



1.2 Save MDF File

```
[7]: from asammdf import MDF
```

```
[8]: # Generated in 90_BasicSignals.ipynb
mdf_file = "90_BasicSignals.mf4"
```

```
[9]: mdf = MDF(
        version='4.11',
    )
mdf.append(
    signals=sigs,
    common_timebase=True,
)
o = mdf.save(
    dst=mdf_file,
    overwrite=True,
    compression=2,
)
```

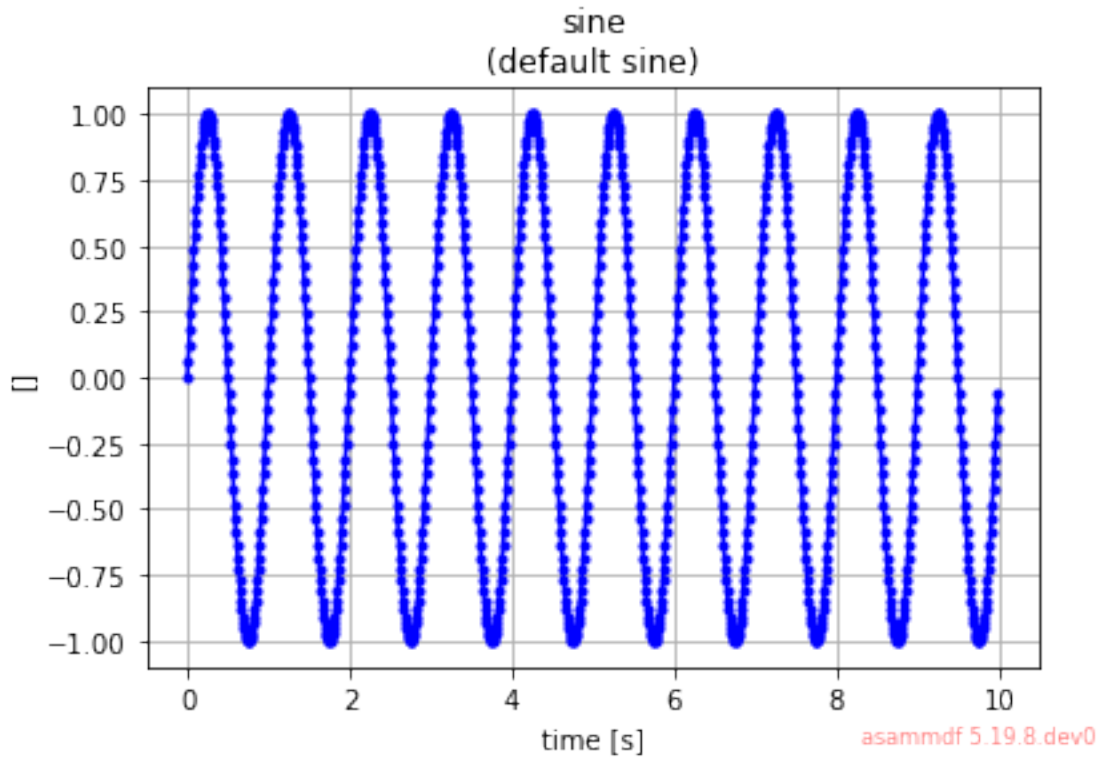
```
[10]: # Clear out the old object.
del mdf
```

2 Open MDF & Plot Signals

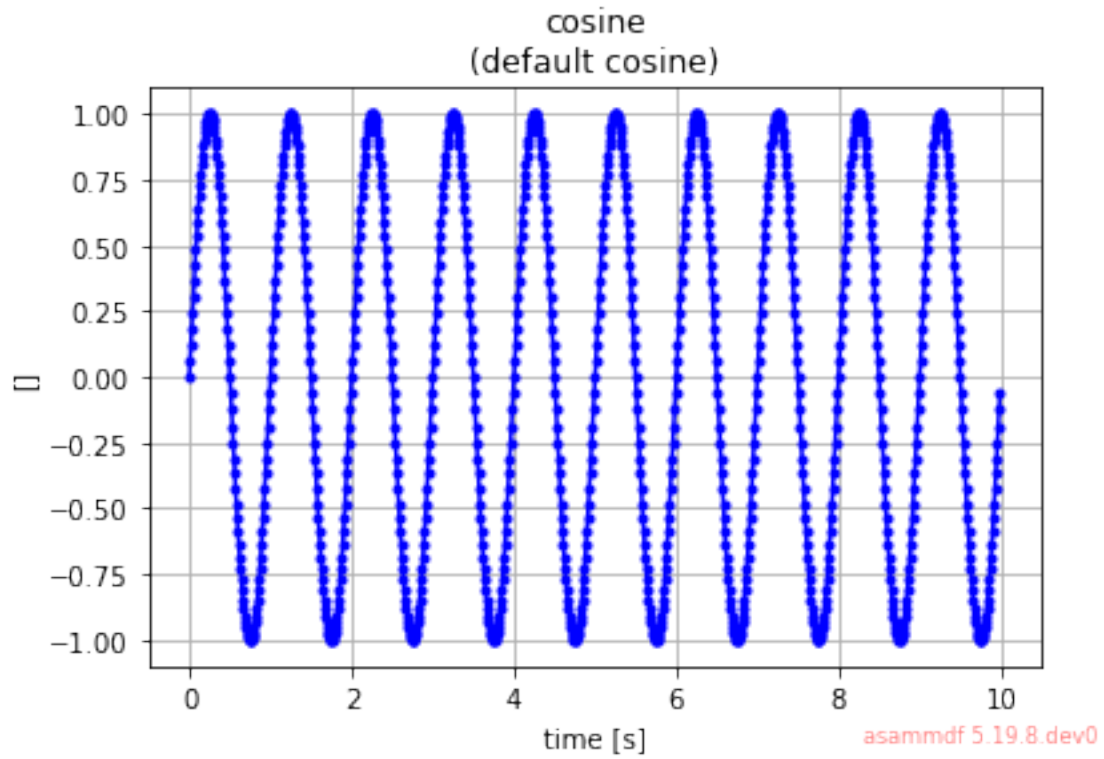
```
[11]: mdf = MDF(mdf_file)
```

```
[12]: for channel in mdf.iter_channels():  
      channel.plot()
```

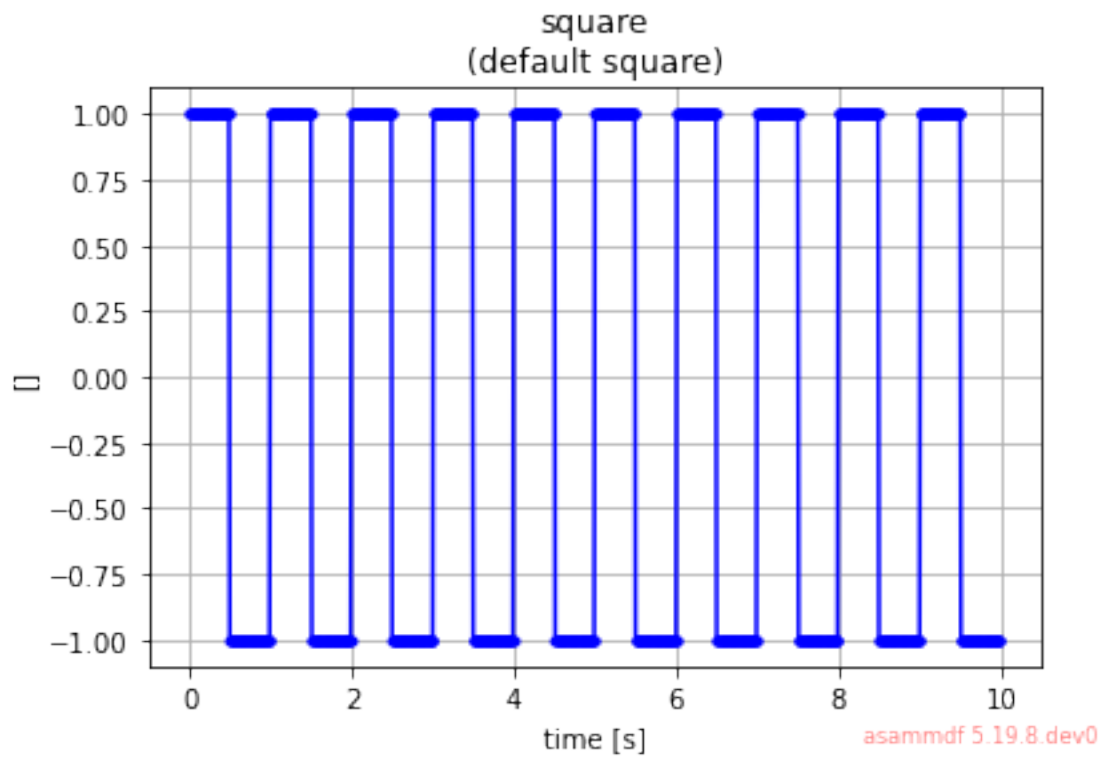
WARNING:root:Signal plotting requires pyqtgraph or matplotlib



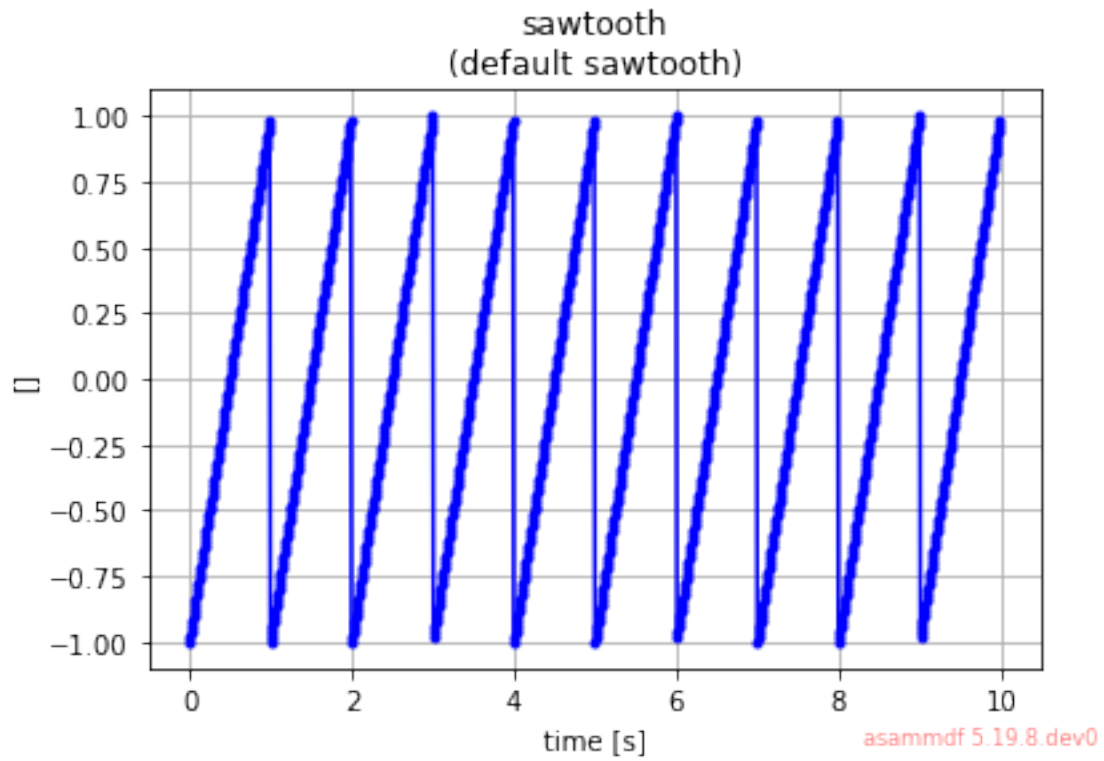
WARNING:root:Signal plotting requires pyqtgraph or matplotlib



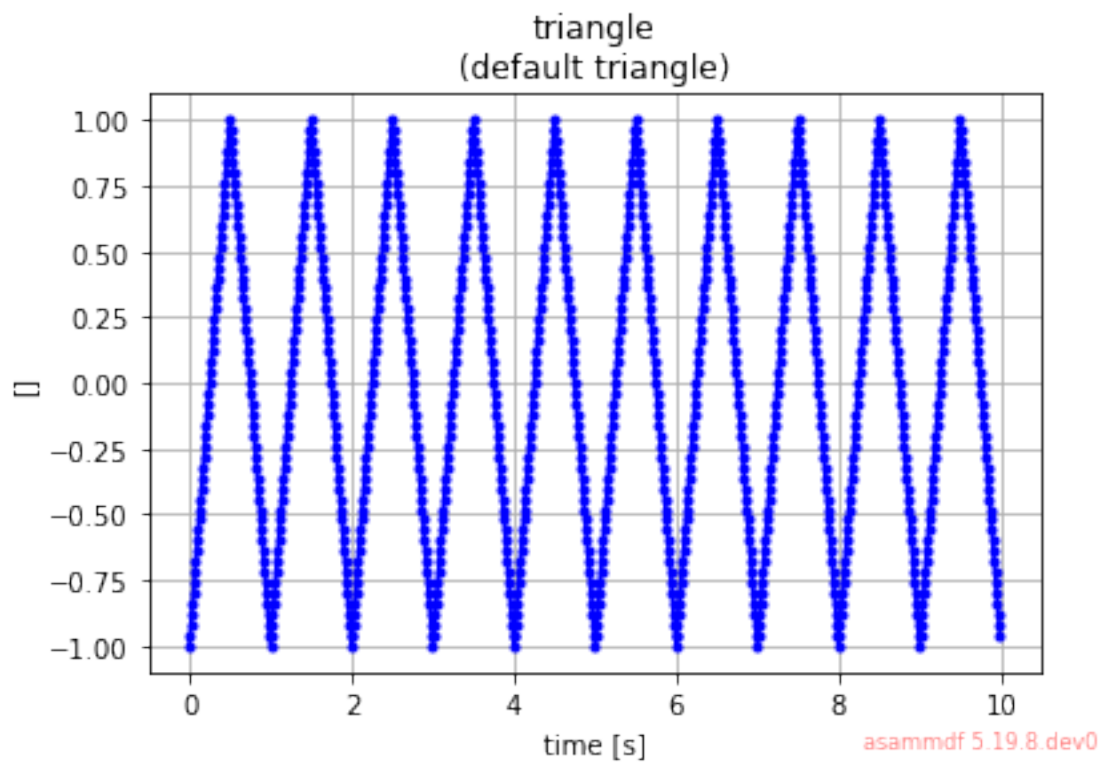
WARNING:root:Signal plotting requires pyqtgraph or matplotlib



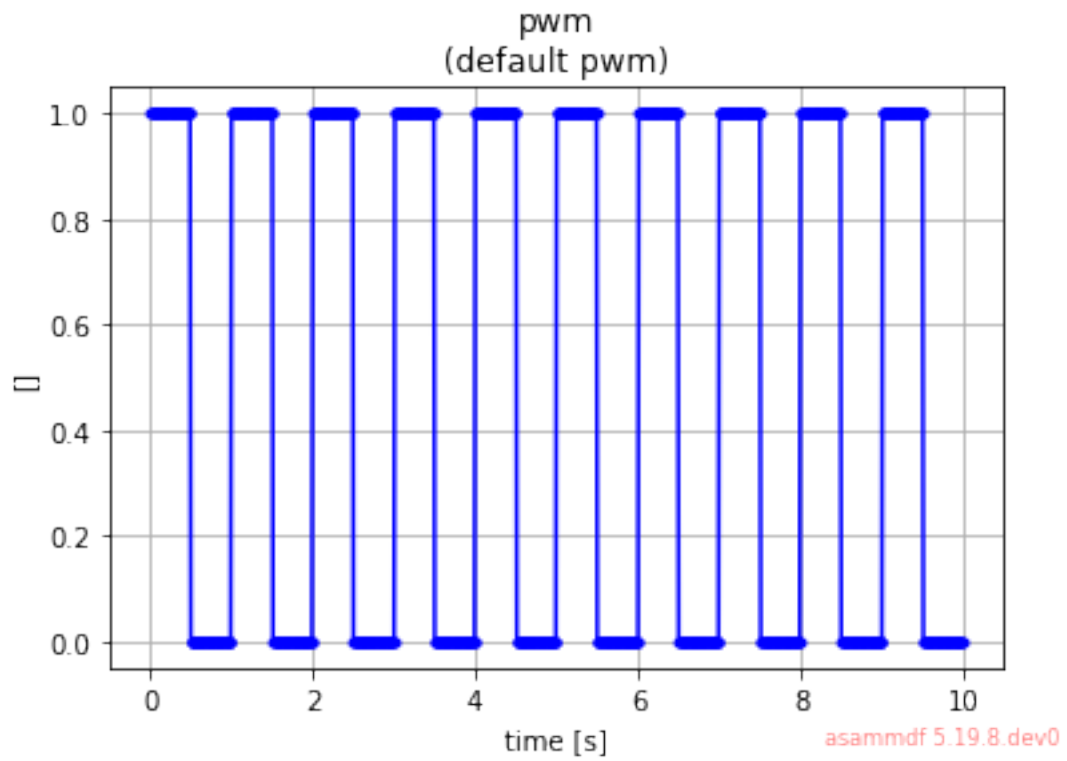
WARNING:root:Signal plotting requires pyqtgraph or matplotlib



WARNING:root:Signal plotting requires pyqtgraph or matplotlib



WARNING:root:Signal plotting requires pyqtgraph or matplotlib



WARNING:root:Signal plotting requires pyqtgraph or matplotlib

