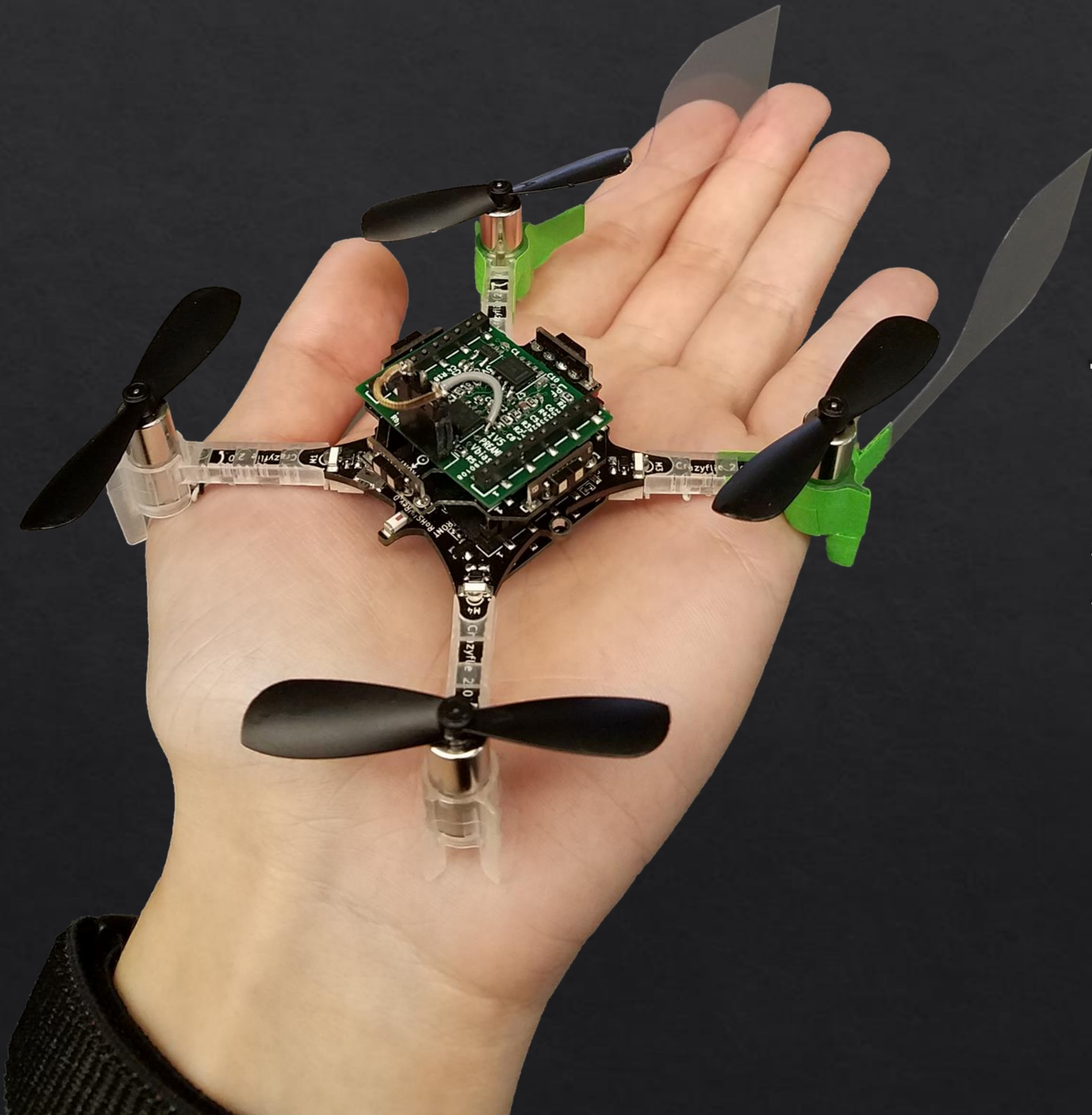


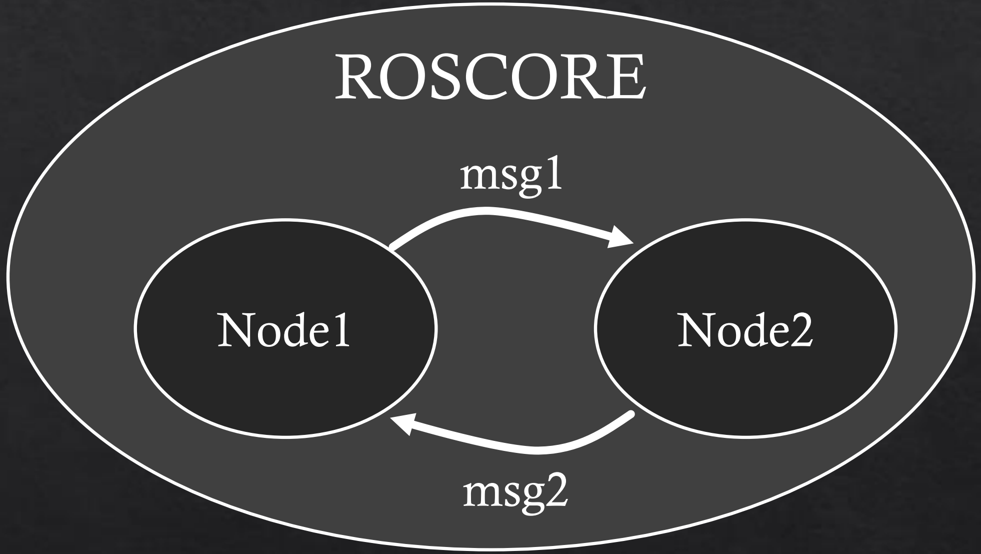
ME586

Introduction to ROS

ROS

- The Robot Operating System (ROS) is a set of software libraries and tools that help you build robot applications.

- What I like about ROS:
  - Easy to write and work with “nodes”
  - Easy to communicate between “nodes”
  - Large user database to debug
  - Portable to different robotic platforms



- This will be a very light lecture on ROS, you can learn more at http://wiki.ros.org/ROS/Tutorials

Why use ROS?

Odor Localization

se ROS?

Odor Localization

Obstacle Avoidance

Common ROS commands

- Create a node
  - Create your python node file in your examples folder (you can use nano or a text editor)
  - Run `chmod +x file_name.py` in order to make it executable by rosrun
  - Then in catkin_ws, run `catkin_make` to rebuild the workspace
- Run the node (either works, just make sure you have a roscore running first!)
  - `python file_name.py`
  - `rosrun rospy_crazyflie file_name.py` (can be used from any directory)

- `rosnode list` will show you the nodes that are currently running
- `rostopic list` will list the current topics being published
- `rostopic echo -c /topic_name` will print out the data being published to the topic (-c clears)
- `rosrun rqt_graph rqt_graph` will show you a depiction of the relationship between the current nodes
- `rosrun rqt_plot rqt_plot` will plot data being published

Example file: Publisher

```
import rospy
from std_msgs.msg import String

def talker():
    pub = rospy.Publisher('chatter', String, queue_size=10)
    rospy.init_node('talker', anonymous=True)
    rate = rospy.Rate(10) # 10hz
    while not rospy.is_shutdown():
        hello_str = "hello world %s" % rospy.get_time()
        rospy.loginfo(hello_str)
        pub.publish(hello_str)
        rate.sleep()

if __name__ == '__main__':
    try:
        talker()
    except rospy.ROSInterruptException:
        pass
```

Example file: Subscriber

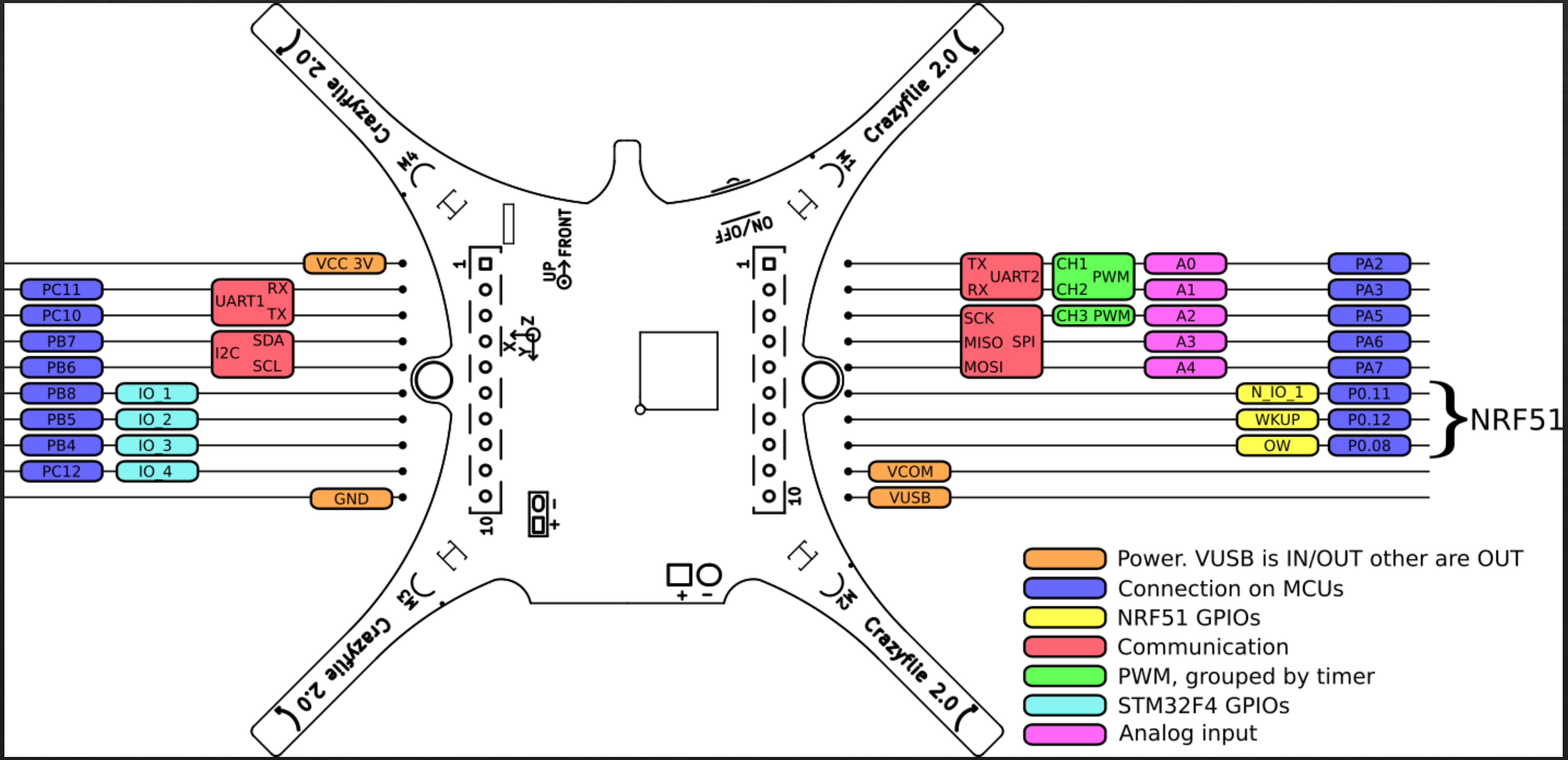
```python
import rospy
from std_msgs.msg import String

def callback(data):
    rospy.loginfo(rospy.get_caller_id() + "I heard %s", data.data)

def listener():
    rospy.init_node('listener', anonymous=True)
    rospy.Subscriber("chatter", String, callback)

    # spin() simply keeps python from exiting until this node is stopped
    rospy.spin()

if __name__ == '__main__':
    listener()
```

Problem set questions

1. Write a node which takes in keyboard input from the user and publishes it to a topic
2. Write a node which subscribes to the keyboard topic, controls the crazyflie, and publishes the flight data to a different topic
3. Write a node which subscribes to the flight data topic and live plots the data
4. Take a short video showing all of these things working together

Adding a sensor to the crazyflie

- Communicating through the crazyflie (more technical stuff about this later)



Adding a sensor to the crazyflie

- Communicating outside the crazyflie