

Diario di lavoro

Luogo: Canobbio

Data: 06.12.2019

Lavori svolti

Oggi ho continuato buona parte della documentazione.

Poi mi sono documentato sul sito: <https://www.palletsprojects.com/p/flask/> per cercare di capire come funziona il webserver flask, per creare il codice dell'acc lato arduino OpenWRT. Ed infine ho iniziato a scrivere, il codice del bridge, cioè un sistema che mi permetta di poter scrivere sul bridge, direttamente da diverse thread, quindi deve essere un codice sincrono.

```
import sys
sys.path.insert(0, '/usr/lib/python2.7/bridge')
from threading import Lock
from bridgeclient import BridgeClient
class BridgeGlobal:
    def __init__(self):
        """
        Create global bridge, initialize the bridge client and the lock
        for synchronization.
        """
        # initialize bridge
        self.bridge = BridgeClient()
        # initialize synchronization lock
        self.lock = Lock()
    def get(self, key):
        """
        Get from the bridge with synchronization.
        :param key: Key of the value (memory address).
        :return: Value of the key.
        """
        # acquire the synchronization lock
        self.lock.acquire()
        # get the value
        value = self.bridge.get(key)
        # release the synchronization lock
        self.lock.release()
        # return the value
        return value
    def put(self, key, value):
        """
        Put the value in the bridge with synchronization.
        :param key: Key of the value (memory address).
        :param value: Value to set.
        """
        # acquire the synchronization lock
```

```
self.lock.acquire()  
# set the value  
self.bridge.put(key, value)  
# release the synchronization lock  
self.lock.release()
```

Problemi riscontrati e soluzioni adottate

-

Punto della situazione rispetto alla pianificazione

Sviluppare il codice per OpenWRT.

Programma di massima per la prossima giornata di lavoro

-