Cryptography has a very important role to play in embedded systems and the Internet of Things, due to the need for these systems to communicate with each other securely. Various algorithms exist for secure random number generation that are both fast and utilize minimal resources. However, the security of these algorithms relies on the randomness of their seed value, these systems typically have limited or non-standard sources of entropy.

To solve this, we propose implementing a Hybrid Random Number Generator (HRNG) on a prototype hardware platform. We will implement the Blum Blum Shub CSRNG as the primary component of the system. This algorithm is known to be slow, so we will investigate optimizations, including Montgomery Exponentiation to increase performance. In order to seed the CSPRNG, we will investigate an algorithm for using sensors and other physical processes on our development platform to create entropy. This will require creating a specific process of obtaining the randomness from each source, and coherently mixing it together to obtain the final seed value.

To evaluate our implementation, we will measure the randomness of our extracted entropy, as well as the randomness of the resulting CSPRNG output, using standard randomness tests. We can also evaluate the bandwidth, to ensure that the system can produce enough output to meet the needs of a prototypical embedded application. The system will be implemented and evaluated on the UDOO development platform, giving us a balance of ease-of-implementation, power, and similarity to a real-world deployment.